МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Гжельский государственный университет»
(ГГУ)
Колледж ГГУ

Специальность 09.02.07 Информационные системы и программирование
Форма обучения – очная

ДИПЛОМНЫЙ ПРОЕКТ

на тему: **«Разработка информационной предприятия оптовой торговли (ООО «Руссоль-центр»)»**

Выполнила:
студентка группы ИСП-О-22
Вербицкая Елена Алексеевна
_____
(подпись)

РАБОТА ДОПУЩЕНА К ЗАЩИТЕ
Заместитель директора по УР
М.В. Казакова
_____
(подпись)

Руководитель:
преподаватель колледжа,
Зимина Анна Сергеевна
_____
(подпись)

МП

пос. Электроизолятор
2025 год

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 5
1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 8
1.1. Описание и анализ предметной области 8
1.2. Принципы разработки ИС 11
1.3. Характеристика проблемы разработки 14
1.4. Обзор и сравнительный анализ готовых решений 17
2. ПРОЕКТНАЯ ЧАСТЬ 23
2.1. Определение требований к информационной системе 23
2.2. Обзор и выбор средств проектирования и разработки ИС 27
2.2.1. Язык программирования и среда разработки 28
2.2.2. Технология интерфейса (GUI Framework) 28
2.2.3. Система управления базами данных (СУБД) 29
2.2.4. Инструменты проектирования и моделирования 29
2.2.5. Система контроля версий 30
2.2.6. Выбранные средства 30
2.3. Методы моделирования системы 31
2.3.1. ER-диаграмма 32
2.3.2. Контекстная диаграмма 35
2.3.3. Диаграмма декомпозиции IDEF0 38
2.3.4. Диаграмма потоков данных 42
2.3.5. Диаграмма вариантов использования (Use Case Diagram) 46
2.4. Макеты интерфейса приложения 50
2.5. Описание логики функционирования ПО 81
2.5.1. Общая архитектура и компоненты ПО 82
2.5.2. Вход в систему и разграничение прав 82
2.5.3. Работа с заказами 83

2.5.4. Работа с оказанными услугами 84
2.5.5. Работа со справочниками и складом 84
3. СПЕЦИАЛЬНАЯ ЧАСТЬ 86
3.1. Формирование требований к ПО 86
3.2. Разработка интерфейса ПО 89
3.3. Проектирование структуры базы данных 101
3.3.1. Физическая ER диаграмма 103
3.3.2. Описание таблиц 103
3.3.3. Связи между таблицами 115
3.3.4. Индексы в БД 119
3.4. Отладка и тестирование программы 121
3.4.1. Цель и задачи отладки и тестирования 121
3.4.2. Методы и виды тестирования 122
3.4.3. Инструменты отладки и тестирования 124
3.4.4. Проведение тестирования 124
3.4.5. Результаты тестирования 126
3.5 Руководство пользователя 127
3.6 Руководство программиста 130
ЗАКЛЮЧЕНИЕ 132
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 134
ПРИЛОЖЕНИЕ 136

ВВЕДЕНИЕ

Современные условия функционирования предприятий оптовой торговли требуют высокого уровня автоматизации бизнес-процессов, эффективного управления ресурсами, оперативного обмена информацией между подразделениями и внешними контрагентами. Информационные системы становятся ключевым инструментом, обеспечивающим прозрачность, точность и скорость выполнения операций, начиная от формирования заказов и заканчивая логистикой и финансовым учетом. Особенно это актуально для специализированных предприятий, таких как ООО «Руссоль-центр», занимающееся оптовой продажей соли и сопутствующих товаров, где сложные логистические цепочки и разнообразие номенклатуры требуют грамотной автоматизации и централизованного контроля.

Актуальность темы выпускной квалификационной работы обусловлена необходимостью повышения эффективности управления и снижения операционных издержек на предприятиях оптовой торговли за счет внедрения современных информационных технологий. Ручная обработка заказов, неэффективное управление складскими запасами и ограниченный контроль над финансовыми потоками значительно замедляют бизнес-процессы и увеличивают вероятность ошибок. Разработка специализированной информационной системы позволяет автоматизировать ключевые функции, сократить временные затраты, улучшить взаимодействие между сотрудниками и повысить общую конкурентоспособность компании.

Объектом исследования в рамках данной работы выступает предприятие оптовой торговли — ООО «Руссоль-центр», включающее его структуру, основные виды деятельности, бизнес-процессы, связанные с управлением заказами, складом, клиентами и поставками.

Предметом исследования являются бизнес-процессы, связанные с деятельностью отдела продаж и складского хозяйства ООО «Руссоль-центр»,

а также возможности их автоматизации и оптимизации с помощью разработки специализированного программного обеспечения.

Целью выпускной квалификационной работы является разработка информационной системы предприятия оптовой торговли на примере ООО «Руссоль-центр», направленной на автоматизацию ключевых бизнес-процессов, таких как управление заказами, складским учетом, взаимодействие с клиентами и поставщиками, а также формирование аналитической отчетности.

Для достижения поставленной цели в рамках работы были определены и решены следующие задачи:

1) Провести анализ предметной области и существующих бизнес-процессов на примере ООО «Руссоль-центр».

2) Выполнить обзор и сравнительный анализ существующих решений и инструментальных средств для проектирования и разработки информационных систем.

3) Сформировать требования к разрабатываемой информационной системе.

4) Спроектировать архитектуру системы, включая диаграммы IDEF0, DFD, Use Case и ER-диаграмму.

5) Разработать макеты пользовательского интерфейса.

6) Осуществить разработку базы данных и программную реализацию отдельных модулей системы.

7) Подготовить документацию и руководство пользователя.

Работа состоит из введения, трех глав, заключения, списка использованных источников и приложений. В первой главе (аналитической) представлен обзор теоретических основ проектирования информационных систем, проведен анализ предметной области и рассмотрены аналогичные решения. Вторая глава (проектная) посвящена выбору инструментов разработки, проектированию архитектуры системы, разработке диаграмм и макетов интерфейса. Третья глава включает в себя разработку базы данных,

описание программной реализации отдельных модулей и результаты тестирования. Обобщающий вывод представлен в заключении.

Таким образом, выпускная квалификационная работа направлена на решение актуальной задачи по автоматизации бизнес-процессов предприятия оптовой торговли, что делает её значимой как с теоретической, так и с практической точки зрения.

1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1. Описание и анализ предметной области

Предметная область — это часть реального мира, рассматриваемая в пределах данного контекста. В рамках дипломной работы под предметной областью понимается совокупность объектов, процессов, понятий, связей и правил, которые относятся к деятельности предприятия оптовой торговли ООО «Руссоль-центр» и являются объектом исследования и автоматизации.

В данном случае предметная область охватывает все аспекты, связанные с функционированием компании: от взаимодействия с поставщиками и клиентами до внутреннего управления складскими запасами, финансовыми потоками и персоналом. Она включает в себя как физические объекты (товары, складские помещения, транспорт), так и абстрактные понятия (заказ, счет, накладная, статус заказа, должность сотрудника) и процессы (оформление заказа, прием товара, отгрузка, расчет стоимости).

Особое внимание в рамках данной работы уделяется бизнес-процессам, связанным с деятельностью отдела продаж и складского хозяйства, поскольку именно они формируют основу для проектирования информационной системы. Анализ предметной области позволяет выявить ключевые операции, участников, входящую и выходящую информацию, а также ограничения и правила, которые должны быть учтены при разработке программного обеспечения.

Для более глубокого понимания структуры и функционирования предприятия, на рисунке 1.1 представлена организационная структура ООО «Руссоль-центр». Эта схема наглядно демонстрирует иерархию управления, распределение полномочий и ответственности между подразделениями, что является важнейшим элементом для определения акторов (пользователей) будущей информационной системы и их ролей.

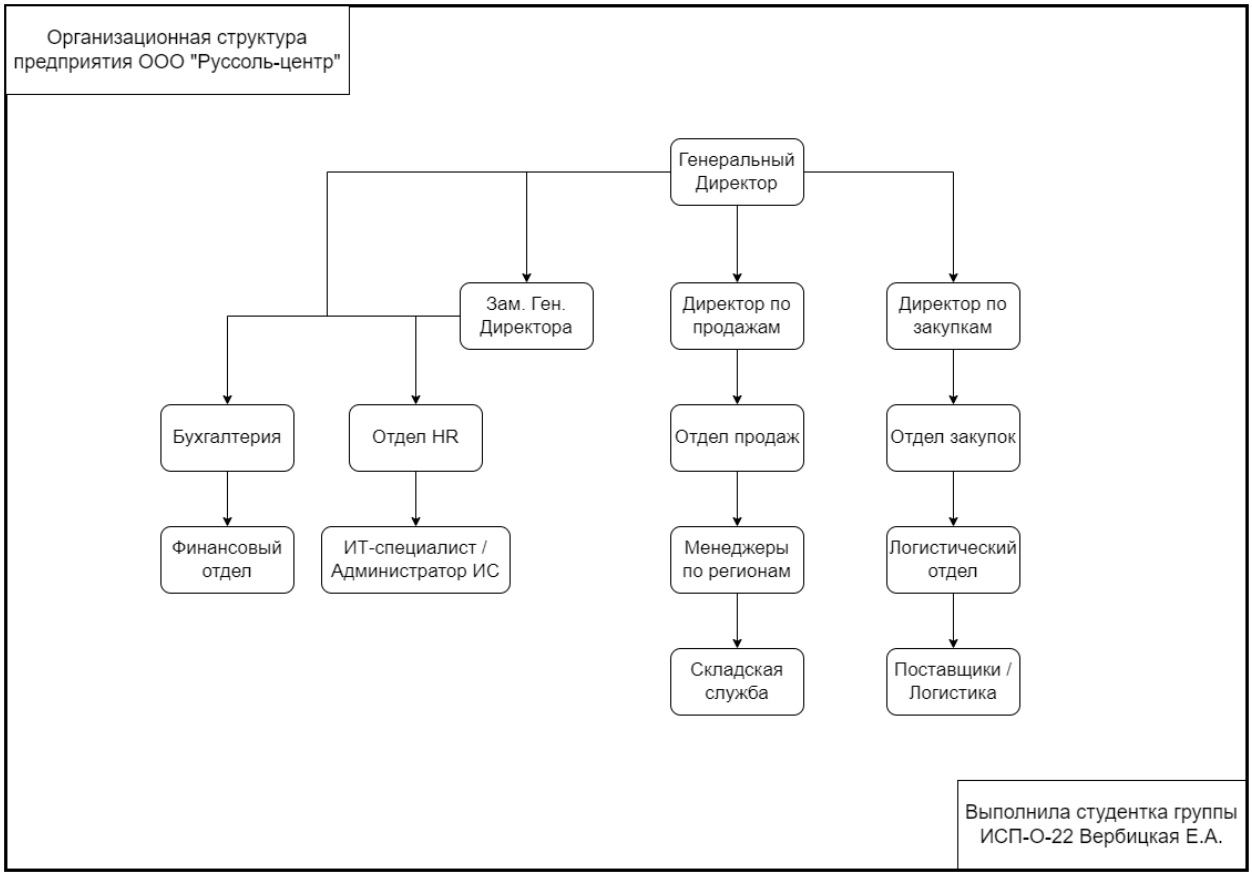

Рисунок 1.1. Организационная структура ООО «Руссоль-центр»

Как видно из схемы, предприятие имеет линейно-функциональную структуру управления, что характерно для многих средних компаний. В вершине иерархии находится Генеральный директор, который несет полную ответственность за все аспекты деятельности компании и принимает ключевые стратегические решения. Непосредственно под ним находятся три ключевых управленческих позиции: Заместитель Генерального директора, Директор по продажам и Директор по закупкам. Такое разделение позволяет обеспечить четкое разграничение зон ответственности.

Под Заместителем Генерального директора выделяются два основных направления: Бухгалтерия и Отдел HR. Бухгалтерия отвечает за финансовый учет, отчетность и контроль затрат, а также управляет Финансовым отделом, который, вероятно, занимается оперативными финансовыми операциями и контролем денежных потоков. Отдел HR (Human Resources) отвечает за подбор персонала, адаптацию, обучение и мотивацию сотрудников. В рамках этого отдела работает ИТ-специалист / Администратор ИС, чья роль в

контексте данной работы является критически важной — он отвечает за функционирование и поддержку всей информационной инфраструктуры предприятия, включая системы учета, связи и документооборота.

Направление продаж возглавляется Директором по продажам, который координирует работу Отдела продаж. Этот отдел, в свою очередь, делится на Менеджеров по регионам, которые работают с клиентами в определенных географических зонах. Непосредственно с выполнением заказов и управлением складскими запасами связаны Складская служба и Логистический отдел. Складская служба отвечает за прием, хранение, комплектацию и отгрузку товаров, а Логистический отдел — за планирование и организацию доставки продукции до конечного потребителя или клиента.

Направление закупок возглавляется Директором по закупкам, который управляет Отделом закупок. Этот отдел отвечает за поиск и выбор поставщиков, ведение переговоров, заключение договоров и контроль качества поступающей продукции. Взаимодействие с внешними поставщиками осуществляется через Поставщиков / Логистику, что указывает на возможное совмещение функций или тесную связь между отделом закупок и логистическим подразделением.

Таким образом, организационная структура ООО «Руссоль-центр» построена таким образом, чтобы обеспечить эффективное разделение труда и управление ключевыми бизнес-процессами: от закупки сырья и материалов до реализации готовой продукции конечным клиентам. Однако существующая структура, особенно в части взаимодействия между отделами продаж, складом и логистикой, предполагает наличие множества точек ручного ввода данных и передачи информации, что может приводить к задержкам, ошибкам и снижению общей эффективности работы. Это создает необходимость в разработке единой информационной системы, которая сможет объединить все эти процессы в единый цифровой поток, обеспечивая прозрачность, скорость и точность обработки информации на всех уровнях управления.

1.2. Принципы разработки ИС

Разработка информационной системы для предприятия оптовой торговли, в частности для ООО «Руссоль-центр», должна строиться на основе ряда общепринятых и специфических принципов, обеспечивающих её эффективность, надежность, удобство использования и долгосрочную перспективу. Эти принципы определяют не только подход к архитектуре и функционалу системы, но и методы её проектирования, разработки, тестирования и сопровождения. Ниже представлены ключевые принципы, которые должны быть учтены при создании ИС.

1) Принцип соответствия требованиям предметной области и заказчика. Информационная система должна быть ориентирована на специфику деятельности ООО «Руссоль-центр». Это включает в себя автоматизацию ключевых бизнес-процессов: управление заказами клиентов, взаимодействие с поставщиками, учёт и управление складскими запасами, учёт финансов и формирование отчётности. Система должна учитывать существующие бизнес-процессы, а также быть гибкой для адаптации к возможным изменениям. Важно, чтобы функциональность системы напрямую соответствовала целям и задачам предприятия, определённым в техническом задании.

2) Принцип модульности и масштабируемости. Архитектура системы должна быть модульной, то есть состоять из отдельных, слабо связанных между собой компонентов (модулей), таких как модуль заказов, модуль склада, модуль клиентов, модуль поставок и т.д. Это обеспечивает:

- Возможность поэтапного внедрения функционала.
- Упрощение сопровождения, отладки и обновления отдельных частей системы.
- Гибкость: возможность добавлять новые модули (например, модуль аналитики, модуль интеграции с внешними сервисами) без кардинального изменения всей системы.
- Масштабируемость: система должна быть способна работать эффективно при увеличении числа пользователей, объёма данных и

количества одновременных операций, не теряя производительности. Это особенно важно при росте объёмов бизнеса.

3) Принцип интеграции с другими системами. Современное предприятие использует различные программные продукты (бухгалтерские программы, CRM-системы, платёжные шлюзы, логистические сервисы). Разрабатываемая ИС должна предусматривать возможность интеграции с уже используемыми или потенциально востребованными внешними системами. Это может быть реализовано через API (Application Programming Interface), обмен файлами данных в стандартных форматах (XML, JSON, CSV) или другие механизмы. Интеграция позволяет избежать дублирования данных и ручного переноса информации, повышая общую эффективность и точность учёта.

4) Принцип удобства и интуитивности пользовательского интерфейса (UI/UX). Система должна быть удобной и понятной для конечных пользователей — сотрудников различных отделов (менеджеров, кладовщиков, бухгалтеров). Это включает:

- Четкую и логичную навигацию.
- Простой и понятный дизайн элементов управления.
- Минимизацию количества действий для выполнения типичных задач.
- Возможность настройки интерфейса под индивидуальные предпочтения или роли пользователя.
- Обеспечение доступности интерфейса для пользователей с разным уровнем подготовки. Применение современных подходов к UX/UI-дизайну позволяет сократить время на обучение и повысить удовлетворённость пользователей.

5) Принцип надежности и безопасности данных. Информационная система хранит критически важные данные компании: информацию о клиентах, поставщиках, товарах, заказах, финансах. Обеспечение безопасности этих данных является приоритетной задачей. Принцип включает:

- Реализацию механизма аутентификации и авторизации пользователей с различными уровнями доступа (ролевая модель).
- Защиту данных от несанкционированного доступа, модификации и уничтожения.
- Обеспечение целостности данных при работе системы.
- Наличие механизмов резервного копирования и возможности восстановления данных в случае сбоев или ошибок.

6) Принцип автоматизации бизнес-процессов. Одной из основных целей внедрения ИС является автоматизация рутинных и трудоёмких операций. Система должна минимизировать ручной ввод данных, автоматизировать процессы формирования заказов, расчёта сумм, обновления остатков на складе, генерации отчётов и уведомлений. Это позволяет сократить время на выполнение задач, снизить вероятность ошибок и высвободить сотрудников для решения более сложных аналитических и стратегических вопросов.

7) Принцип кроссплатформенности и доступности. В зависимости от требований и инфраструктуры предприятия, ИС может быть реализована как веб-приложение, десктопное приложение или гибридное решение. Веб-ориентированность обеспечивает доступ к системе с различных устройств (ПК, ноутбуки) и операционных систем (Windows, Linux, macOS), при условии наличия браузера и доступа к серверу. Это повышает доступность системы для сотрудников и упрощает развертывание.

8) Принцип соответствия стандартам и нормативам. Разработка ИС должна учитывать применимые законодательные и отраслевые нормы, включая требования по защите персональных данных, бухгалтерскому учёту и электронному документообороту. Система может быть спроектирована с учётом общепринятых стандартов проектирования программного обеспечения (например, UML для моделирования) и стандартов данных.

9) Принцип экономической целесообразности. Система должна быть не только функциональной, но и экономически обоснованной. Это подразумевает:

- Использование рациональных и не требовательных к ресурсам технологий.
- Учёт стоимости разработки, внедрения и сопровождения.
- Обеспечение окупаемости инвестиций за счёт повышения эффективности бизнеса.
- Возможность поэтапного внедрения для распределения затрат.

Следование этим принципам позволяет создать качественную, устойчивую и полезную информационную систему, которая действительно будет способствовать оптимизации деятельности ООО «Руссоль-центр» и повышению её конкурентоспособности.

1.3. Характеристика проблемы разработки

Разработка информационной системы для предприятия оптовой торговли, на примере, представляет собой комплексную задачу, решение которой направлено на повышение эффективности бизнес-процессов, автоматизацию рутинных операций и улучшение качества управления. Однако, несмотря на очевидную пользу, реализация такого проекта сталкивается с рядом специфических проблем, которые необходимо учитывать на всех этапах — от анализа требований до внедрения и сопровождения. Ниже приведён подробный анализ ключевых проблем, возникающих при разработке ИС для данного типа предприятий.

1) Проблема неэффективности управления данными и документооборотом.

Одной из основных проблем, с которой сталкиваются предприятия оптовой торговли, является разрозненность информации. Данные о клиентах, поставщиках, товарах, заказах, остатках на складе, ценах, скидках, финансах могут храниться в различных форматах и местах: в Excel-таблицах, бумажных регистрах, на локальных компьютерах сотрудников, в электронной почте. Это приводит к целому ряду негативных последствий: трудности при поиске актуальной информации, риск дублирования данных, потеря или искажение информации, увеличение времени на выполнение операций, снижение

точности отчётности. Отсутствие единого цифрового информационного пространства затрудняет принятие обоснованных управленческих решений и снижает общую прозрачность бизнес-процессов.

2) Проблема отсутствия автоматизации ключевых бизнес-процессов.

Многие операции в оптовой торговле до сих пор выполняются вручную или с минимальной автоматизацией. Это может включать в себя: формирование заказов клиентов, резервирование товара на складе, проверку наличия, формирование накладных, расчёт стоимости заказа, ведение переговоров с поставщиками, формирование заявок на закупку, контроль сроков поставки, начисление скидок, учёт оплат. Такой подход требует значительных временных и человеческих ресурсов, повышает вероятность ошибок (например, перепутанная номенклатура в заказе, неправильная цена, несвоевременная отгрузка) и снижает общую эффективность. Информационная система должна минимизировать ручной труд, автоматизировать рутинные задачи, устанавливать чёткие бизнес-правила и логику взаимодействия между различными процессами.

3) Проблема несогласованности и отсутствия прозрачности между подразделениями.

В процессе выполнения заказа задействованы различные отделы: отдел продаж (менеджеры), складская служба, логистика, бухгалтерия. При отсутствии единой ИС эти подразделения могут работать с разными источниками информации, не всегда знать о текущем статусе заказа, возникающих проблемах или изменениях. Это приводит к задержкам, недопониманию, конфликтам интересов и снижению качества обслуживания клиентов. Система должна обеспечивать прозрачность процессов, позволяя каждому участнику видеть актуальное состояние заказа, своевременно получать уведомления и выполнять свои функции в установленной последовательности.

4) Проблема управления складскими запасами.

Эффективное управление складом — критически важный аспект оптовой торговли. Проблемы возникают, когда сложно получить точную и актуальную информацию об остатках, скорости оборота товаров, сроках годности, размещении на складе. Это может привести к дефициту популярных позиций и, как следствие, упущенным продажам, или к избыточному складированию медленно оборачиваемых товаров, что влечёт за собой дополнительные затраты. Информационная система должна обеспечивать точный учёт остатков в реальном времени, поддерживать функции резервирования, списания, инвентаризации, а также предоставлять аналитику для планирования закупок и оптимизации ассортимента.

5) Проблема аналитики и отчётности.

Для принятия стратегических и оперативных решений руководству и менеджерам необходимы точные и своевременные отчёты. Однако при ручной или частично автоматизированной системе сбор данных для аналитики становится трудоёмким и длительным процессом. Существующие отчёты могут быть неточными, несвоевременными или не отражать нужные метрики (например, прибыльность по клиентам, поставщикам, номенклатуре, анализ эффективности продаж, анализ закупок). ИС должна включать в себя мощный модуль отчётов и аналитики, позволяющий формировать как стандартные, так и настраиваемые отчёты по различным параметрам.

6) Проблема безопасности и целостности данных.

Информационная система будет содержать конфиденциальные данные: персональные данные клиентов, сведения о поставщиках, коммерческие условия, финансовые данные. Утечка или потеря таких данных может нанести серьёзный ущерб репутации и привести к юридическим последствиям. Также важна защита данных от несанкционированного изменения или удаления. Необходимо реализовать надёжную систему аутентификации и авторизации пользователей, разграничение прав доступа, логирование действий, а также механизмы резервного копирования и восстановления данных.

7) Проблема интеграции с внешними системами.

Предприятие может использовать стороннее программное обеспечение: бухгалтерские программы (например, 1С), CRM-системы, платёжные шлюзы, системы электронной отчётности (например, для работы с маркировкой), логистические сервисы. Отсутствие или слабая интеграция между ИС и этими внешними системами приводит к необходимости ручного переноса данных, что увеличивает вероятность ошибок и снижает эффективность. Разрабатываемая система должна предусматривать возможность интеграции через API или другие стандартные протоколы обмена.

8) Проблема сопротивления изменениям и обучения персонала.

Внедрение новой информационной системы всегда связано с изменением привычных рабочих процессов. Сотрудники могут сопротивляться внедрению, если не понимают её пользы или испытывают трудности с освоением. Поэтому важной задачей является разработка интуитивно понятного интерфейса, проведение обучающих мероприятий, создание подробной документации (руководства пользователя) и обеспечение технической поддержки на этапе перехода.

Таким образом, разработка информационной системы для ООО «Руссоль-центр» требует комплексного подхода к решению перечисленных проблем. Успешное преодоление этих трудностей позволит создать эффективную, надёжную и удобную ИС, которая станет мощным инструментом для автоматизации и оптимизации деятельности предприятия.

1.4. Обзор и сравнительный анализ готовых решений

Для обоснованного выбора архитектурного и технологического решения при разработке информационной системы предприятия оптовой торговли (на примере ООО «Руссоль-центр»), необходимо провести анализ существующих программных продуктов, способных удовлетворить основным требованиям проекта. Сравнение будет производиться по критериям, важным для предприятия: функциональность, масштабируемость, интеграция, безопасность, стоимость и простота использования. Это позволит обосновать

необходимость разработки собственного решения или, при возможности, адаптации готового.

1) Общие ERP-системы (например, SAP ERP, 1C:ERP, Oracle NetSuite)

Назначение: Комплексные решения, охватывающие все основные бизнес-процессы предприятия: финансы, логистика, закупки, продажи, управление персоналом, производство.

Преимущества:

- Полнота функционала: охватывают большинство аспектов управления предприятием.
- Интеграция: Данные из разных отделов хранятся в единой базе, обеспечивая прозрачность и согласованность.
- Стандартизация: помогают внедрить унифицированные бизнес-процессы.
- Отчетность: Мощные инструменты аналитики и отчётов.
- Поддержка: Профессиональная техническая поддержка и регулярные обновления.

Недостатки:

- Высокая стоимость: Значительные затраты на лицензии, внедрение, настройку и сопровождение.
- Сложность: требуют длительного периода внедрения и сложной настройки под специфику конкретного бизнеса.
- Избыточность: для специализированного предприятия, сосредоточенного, например, на оптовой продаже соли, функционал может быть избыточным (например, модули управления производством).
- Гибкость: сложно и дорого вносить кардинальные изменения в стандартную функциональность.

Вывод: Для ООО «Руссоль-центр» внедрение полноценной ERP-системы уровня SAP или Oracle, вероятно, будет избыточным и экономически нецелесообразным. Система 1C:ERP, адаптированная под российские реалии, может быть рассмотрена как потенциальный кандидат, но требует детального

анализа соответствия её модулей специфике оптовой торговли солью и сопутствующими товарами, а также оценки сложности и стоимости адаптации.

2) Специализированные CRM/Складские системы (например, RetailCRM, iiko, WMS-системы)

Назначение: Системы, сфокусированные на определённой области: управление взаимоотношениями с клиентами (CRM), ресторанном бизнесе (iiko) или складской логистике (WMS).

Преимущества:

- Специализация: Глубокий функционал в своей области (например, точное управление остатками, автоматизация складских операций, аналитика продаж).
- Простота внедрения: часто проще и быстрее внедрить, чем ERP.
- Гибкость (в рамках своей сферы): многие из них предлагают настройку под нужды бизнеса.
- Интеграция: часто имеют готовые API и интеграции с популярными платёжными системами, доставкой, маркетплейсами.

Недостатки:

- Фрагментация: для полного охвата бизнес-процессов может потребоваться использовать несколько специализированных систем, что усложняет интеграцию и управление.
- Ограниченная функциональность: не покрывают, например, бухгалтерский учёт или закупки, если не интегрированы с другими системами.
- Стоимость интеграции: Подключение нескольких систем может быть затратным.

Вывод: Специализированные решения, такие как CRM или WMS, могут быть полезны для автоматизации отдельных процессов (например, взаимодействие с клиентами или складской учёт). Однако для создания единой ИС, охватывающей все ключевые аспекты деятельности ООО «Руссоль-центр», потребуется либо сложная интеграция нескольких систем, либо выбор

более универсального, но всё ещё специализированного под оптовую торговлю, решения.

3) Open Source решения (например, Odoo, ERPNext, Magento для B2B)

Назначение: Открытые корпоративные системы, часто модульные, с возможностью самостоятельной установки и настройки.

Преимущества:

- Сниженная стоимость лицензии: Основной код бесплатен, что снижает первоначальные затраты.
- Гибкость: Возможность модифицировать код под конкретные нужды (требует квалифицированных разработчиков).
- Сообщество: Большое сообщество пользователей и разработчиков, большое количество готовых модулей.

Недостатки:

- Сложность настройки и разработки: требует значительных знаний и усилий для настройки под сложный бизнес-процесс.
- Отсутствие гарантий: нет официальной поддержки от производителя, только сообщество и/или найм сторонних специалистов.
- Скрытые расходы: Затраты на установку, настройку, интеграцию, сопровождение и разработку могут быть высоки.
- Безопасность: Необходимость самостоятельного обеспечения безопасности и обновлений.

Вывод: Open Source решения могут быть рассмотрены как основа для разработки, особенно если у компании есть внутренние IT-ресурсы. Однако для ООО «Руссоль-центр», вероятно, более целесообразным будет либо использование облачной версии (SaaS) таких систем с ограниченной кастомизацией, либо разработка собственного решения, чтобы избежать сложностей с адаптацией чужой архитектуры под специфику.

4) Облачные SaaS-решения (Software as a Service) (например, amoCRM + WMS, Bitrix24 с модулями, Microsoft Dynamics 365 Business Central)

Назначение: Программное обеспечение, размещенное в облаке и доступное по подписке. Может быть как универсальным, так и специализированным.

Преимущества:

- Минимальные начальные затраты: нет необходимости в собственном серверном оборудовании и лицензиях.
- Простота запуска: Быстрое развертывание.
- Обновления: Автоматические обновления функционала и безопасности.
- Масштабируемость: легко масштабируется в зависимости от роста бизнеса.
- Доступность: Доступ к системе из любого места с интернетом.

Недостатки:

- Зависимость от провайдера: Управление системой и данными частично находится в руках поставщика.
- Ограниченная кастомизация: Возможности настройки под очень специфические нужды ограничены.
- Ежемесячные расходы: Расходы на подписку могут со временем превысить стоимость одноразовой покупки.
- Безопасность данных: Необходимость доверия провайдеру хранения и обработки конфиденциальной информации.

Вывод: SaaS-решения могут быть хорошим выбором для быстрого старта и при отсутствии собственных IT-ресурсов. Однако, при стремлении к полной автоматизации и контролю над всеми процессами, как в случае ООО «Руссоль-центр», стандартные SaaS-продукты могут не обеспечить необходимой гибкости и интеграции.

Заключение по анализу:

Проведённый анализ показывает, что универсального "коробочного" решения, идеально подходящего под специфику деятельности ООО «Руссоль-центр» (специализированное предприятие оптовой торговли, с узкой

номенклатурой, специфическими клиентами и поставщиками), не существует. Стандартные ERP-системы избыточны и дороги. Специализированные решения решают отдельные задачи, но создают фрагментацию. Open Source и SaaS решения требуют либо высокой квалификации, либо компромиссов в функциональности.

Таким образом, наиболее целесообразным представляется разработка собственной информационной системы, спроектированной с нуля под конкретные бизнес-процессы ООО «Руссоль-центр». Это позволит:

- Создать гибкое и масштабируемое решение.
- Обеспечить полную интеграцию всех ключевых процессов.
- Учесть все специфические требования бизнеса.
- Контролировать безопасность и целостность данных.
- Добиться оптимального соотношения функциональности и стоимости в долгосрочной перспективе, не переплачивая за ненужные модули или чужую инфраструктуру.

2. ПРОЕКТНАЯ ЧАСТЬ

2.1. Определение требований к информационной системе

Определение требований к информационной системе (ИС) является критически важным этапом в процессе её разработки. Этот этап включает в себя сбор, анализ, документирование и согласование требований, которые система должна выполнить для удовлетворения потребностей пользователей, бизнеса и других заинтересованных сторон. От точности и полноты сформулированных требований зависит успех всего проекта: архитектура системы, функциональность, удобство использования, стоимость разработки и сроки внедрения. В рамках дипломной работы для ООО «Руссоль-центр» требования формулируются с учётом анализа предметной области, выявленных проблем и специфики деятельности предприятия.

Общие цели ИС

Основной целью разрабатываемой информационной системы является автоматизация и оптимизация ключевых бизнес-процессов ООО «Руссоль-центр», связанных с оптовой торговлей. Это включает в себя:

- Повышение оперативности и точности обработки заказов клиентов.
- Эффективное управление складскими запасами.
- Улучшение взаимодействия между отделом продаж, складской службой и логистикой.
- Автоматизацию учёта и анализа финансовой и товарной информации.
- Обеспечение централизованного хранения данных о клиентах, поставщиках, товарах и операциях.
- Повышение удовлетворённости клиентов за счёт более точного и быстрого обслуживания.

Категории пользователей (акторы)

В системе можно выделить несколько категорий пользователей, каждый из которых будет взаимодействовать с системой по-своему, в зависимости от своих ролей и обязанностей:

- Администратор: Пользователь с максимальными правами доступа. Отвечает за общее администрирование системы, включая управление пользователями, настройки системы, доступ к полному функционалу, а также контроль за корректностью данных и работой системы. Имеет возможность просматривать, создавать, редактировать и удалять любые данные (клиенты, заказы, услуги, товары, сотрудники…).
- Менеджер по заказам: Основной пользователь, работающий с заказами клиентов, клиентской информацией, товарами, ценами и скидками. Должен иметь возможность создавать, просматривать, редактировать и отслеживать статусы заказов. Может работать с информацией о клиентах и товарах.
- Менеджер по услугам: Пользователь, ответственный за учёт и оформление оказанных клиентам услуг. Может создавать, просматривать и изменять записи об оказанных услугах, привязывая их к клиентам и сотрудникам. Может просматривать информацию о клиентах и сотрудниках, но не может управлять ими.

Функциональные требования

Функциональные требования описывают, что система должна делать, какие функции и операции она должна выполнять.

Управление клиентами:

F1.1. Возможность добавления, редактирования и просмотра информации о клиентах (название, контактное лицо, телефон, email, адрес, юридические данные).

F1.2. Возможность поиска и фильтрации клиентов по различным параметрам (название, регион, тип клиента).

F1.3. Ведение истории взаимодействия с клиентом (сделки, заказы, комментарии).

Управление поставками и поставщиками:

F2.1. Управление справочником поставщиков (название, контакты, реквизиты).

F2.2. Фиксация поступления товара от поставщиков (документы, накладные).

F2.3. Отслеживание статуса поставки.

Управление товарами:

F3.1. Ведение справочника товаров (название, артикул, категория, описание, единица измерения, цена).

F3.2. Возможность поиска и фильтрации товаров.

F3.3. Управление ценами и скидками (например, по клиенту, по объёму заказа).

Управление заказами:

F4.1. Создание нового заказа (выбор клиента, добавление позиций товара с указанием количества, применение скидок).

F4.2. Редактирование заказа (до определённого статуса, например, до отгрузки).

F4.3. Просмотр списка заказов с фильтрацией и сортировкой по статусу, клиенту, дате.

F4.4. Изменение статуса заказа (Новый, Подтверждён, В сборке, Отгружен, Доставлен, Отменён).

Управление складом:

F5.1. Отображение актуальных остатков по товарам (по складским ячейкам, если применимо).

F5.2. Формирование заданий на сборку заказов для складских работников.

F5.3. Учёт операций поступления и отпуска товара (в т.ч. внутренние перемещения).

F5.4. Возможность проведения инвентаризации.

Финансовый учёт и отчёты:

F6.1. Фиксация оплат по заказам.

F6.2. Формирование отчётов: по продажам (по периоду, менеджеру, клиенту, товару), по клиентам (задолженность, обороты), по поставкам, по остаткам.

F6.3. Возможность экспорта отчётов в стандартные форматы (Excel, PDF).

Управление пользователями и безопасностью:

F7.1. Регистрация и авторизация пользователей.

F7.2. Назначение ролей и разграничение прав доступа к функциям и данным системы (например, менеджер не может редактировать цены, бухгалтер не может создавать заказы от имени другого менеджера).

Нефункциональные требования

Нефункциональные требования описывают как система должна работать, определяя её характеристики, не связанные напрямую с конкретными функциями.

Производительность:

NF1.1. Система должна отвечать на типичные запросы (просмотр списка заказов, поиск товара) в течение 1-2 секунд при одновременной работе до 10 пользователей.

NF1.2. Загрузка отчётов не должна занимать более 5-10 секунд.

Надёжность:

NF2.1. Система должна быть устойчива к сбоям и обеспечивать целостность данных при корректной эксплуатации.

NF2.2. Обязательно наличие механизма резервного копирования базы данных (ежедневно, возможно, с хранением на внешнем носителе или в облаке).

Безопасность:

NF3.1. Пароли пользователей должны храниться в зашифрованном виде.

NF3.2. Доступ к системе должен происходить только после аутентификации.

NF3.3. Данные при передаче между клиентом и сервером (если применимо) должны быть защищены (например, с использованием SSL/TLS).

Удобство использования (Usability):

NF4.1. Интерфейс должен быть интуитивно понятным и соответствовать общепринятым стандартам (GUI guidelines).

NF4.2. Система должна предоставлять подсказки и сообщения об ошибках на понятном пользователю языке.

NF4.3. Интерфейс должен быть локализован на русский язык.

Совместимость:

NF5.1. Система должна быть совместима с операционной системой, используемой на рабочих станциях ООО «Руссоль-центр» (например, Windows 10/11).

NF5.2. (Опционально) Возможность интеграции с бухгалтерской системой 1С.

Масштабируемость:

NF6.1. Архитектура системы должна предусматривать возможность добавления новых функций и увеличения числа пользователей в будущем.

Сформулированные функциональные и нефункциональные требования определяют, *что* должна из себя представлять информационная система для ООО «Руссоль-центр». Они служат основой для последующих этапов: проектирования архитектуры, разработки базы данных, создания интерфейсов и программного кода. Чёткое определение требований позволяет избежать недоразумений между разработчиками и заказчиком и гарантирует, что финальный продукт будет соответствовать целям и задачам автоматизации, поставленным перед проектом.

2.2. Обзор и выбор средств проектирования и разработки ИС

Для успешной реализации информационной системы для ООО «Руссоль-центр» необходимо выбрать оптимальные инструментальные средства, охватывающие все этапы: от проектирования архитектуры и моделирования данных до написания кода, отладки и тестирования. В этом

разделе рассматриваются ключевые категории инструментов, анализируются различные варианты и обосновывается выбор конкретных технологий и программных продуктов, используемых в проекте.

2.2.1. Язык программирования и среда разработки

Язык программирования и среда разработки (IDE) являются фундаментом для создания приложения. Для десктопного приложения под Windows, особенно в среде Microsoft, приоритет отдается технологиям .NET и C#.

Язык программирования:

- C# (C-sharp): Современный, объектно-ориентированный язык, разработанный Microsoft. Обладает высокой производительностью, строгой типизацией, встроенной поддержкой многопоточности и интеграцией с .NET Framework/.NET Core. Отлично подходит для создания надёжных и масштабируемых корпоративных приложений.
- Альтернативы: Java (платформонезависимость, но сложнее интеграция с Windows), Python (простота, но потенциально ниже производительность для десктопных приложений).

Среда разработки (IDE):

- Microsoft Visual Studio: Мощная и профессиональная IDE, официально поддерживаемая Microsoft для разработки на C#. Предоставляет встроенный отладчик, IntelliSense, средства проектирования интерфейса, интеграцию с Git и SQL Server. Бесплатная Community-версия покрывает потребности большинства проектов.
- Альтернативы: JetBrains Rider (отличная альтернатива, но коммерческая), Visual Studio Code (легковесный редактор, требует плагинов для полноценной C# разработки).

2.2.2. Технология интерфейса (GUI Framework)

Для создания десктопного интерфейса в .NET доступны два основных фреймворка:

- Windows Forms: Простой и проверенный способ создания интерфейсов с привычным внешним видом для Windows. Подходит для приложений с умеренными требованиями к интерактивности и дизайну. Быстрее осваивается.
- WPF (Windows Presentation Foundation): более современная и гибкая технология, позволяющая создавать сложные и визуально богатые интерфейсы с использованием XAML и архитектуры MVVM. Требует больше времени на изучение, но предоставляет большие возможности для кастомизации.
- Альтернативы: .NET MAUI (новейшая кроссплатформенная технология Microsoft, но менее зрелая для десктопа под Windows).

2.2.3. Система управления базами данных (СУБД)

Для хранения и управления данными выбирается СУБД, обеспечивающая надёжность, производительность и удобство администрирования.

- Microsoft SQL Server: Ведущая корпоративная СУБД, разработанная Microsoft. Отлично интегрируется с .NET/C#, обладает высокой производительностью, надёжностью, развитыми средствами безопасности и анализа. Поддерживает сложные запросы и транзакции. Express-версия бесплатна для ограниченного использования.
- Альтернативы: PostgreSQL (мощная open-source СУБД), MySQL (популярна, но может уступать PostgreSQL в сложных задачах), SQLite (легковесная файловая СУБД, не подходит для многопользовательских систем).

2.2.4. Инструменты проектирования и моделирования

Для визуализации архитектуры, данных, процессов и интерфейсов используются специализированные инструменты.

Для диаграмм и моделирования:

- draw.io (diagrams.net): Бесплатный, кроссплатформенный инструмент. Подходит для создания ER-диаграмм, DFD, IDEF0, Use Case и других. Прост в использовании, поддерживает экспорт.
- Microsoft Visio: Профессиональный инструмент с широкими возможностями, но коммерческий.
- Pencil Project: Удобен для создания макетов интерфейсов (mockups).
- Альтернативы: Lucidchart (веб-инструмент с совместной работой), StarUML, Enterprise Architect (мощные CASE-средства, часто коммерческие).

Для управления проектом:

- Microsoft Project: Инструмент для планирования, распределения ресурсов и контроля выполнения задач.
- Альтернативы: Trello, Jira, Asana (онлайн-доски и трекеры задач), GanttProject (бесплатный планировщик).

2.2.5. Система контроля версий

Необходима для отслеживания изменений в коде и совместной работы.

- Git: Стандарт де-факто в современной разработке. Распределённая система, мощные возможности слияния изменений, интеграция с IDE и платформами хостинга кода (GitHub, GitLab).

2.2.6. Выбранные средства

На основе анализа были выбраны следующие средства для разработки информационной системы:

Язык программирования: C# — выбран как основной язык разработки благодаря своей производительности, надёжности, интеграции с .NET и экосистемой Microsoft, что соответствует целевой платформе Windows и потенциальной инфраструктуре предприятия.

Среда разработки: Microsoft Visual Studio Community — выбрана как наиболее полная и интегрированная среда разработки для C# и .NET, обеспечивающая все необходимые инструменты (редактор, отладчик, интеграция с Git и SQL Server) и являющаяся бесплатной для некоммерческого использования.

Технология интерфейса: Windows Forms — выбрана как более простая и быстрая в реализации для создания функционального десктопного интерфейса с привычным внешним видом для пользователей Windows. Позволяет сосредоточиться на логике приложения.

СУБД: Microsoft SQL Server (Express Edition) — выбрана благодаря высокой производительности, надёжности, безопасности и отличной интеграции с .NET/C#. Express-версия удовлетворяет требованиям проекта по объёму данных и функциональности на начальном этапе.

Инструменты моделирования: draw.io (diagrams.net) для создания диаграмм (ER, DFD, IDEF0, Use Case) и Pencil Project для макетов интерфейса — выбраны как бесплатные, доступные и удобные инструменты для визуализации архитектуры и дизайна.

Система контроля версий: Git — выбрана как стандартная и мощная система для отслеживания изменений в коде.

Этот набор технологий обеспечивает:

- Высокую производительность и надёжность приложения.
- Хорошую интеграцию между компонентами (C#, .NET, SQL Server, Visual Studio).
- Совместимость с предполагаемой инфраструктурой предприятия.
- Доступность и низкие начальные затраты (использование бесплатных/условно-бесплатных инструментов).
- Надёжность, масштабируемость и возможности сопровождения решения.

Выбранные инструменты позволяют реализовать все поставленные функциональные и нефункциональные требования к системе и создать эффективное и удобное в сопровождении программное обеспечение для ООО «Руссоль-центр».

2.3. Методы моделирования системы

Моделирование системы — это процесс создания упрощённого представления реальной системы с целью её анализа, проектирования и

разработки. Для формального описания различных аспектов информационной системы, таких как структура данных, функции, процессы и взаимодействие с пользователями, применяются различные методы и нотации. В рамках проекта разработки информационной системы для ООО «Руссоль-центр» используются следующие методы моделирования, реализованные в виде диаграмм. Эти диаграммы позволяют структурировать и формализовать систему, обеспечивая единое понимание между разработчиками и заказчиком.

2.3.1. ER-диаграмма

ER-диаграмма (диаграмма "Сущность-Связь") — это графическая модель, описывающая структуру данных информационной системы. Она показывает сущности (объекты реального мира, например, "Клиент", "Товар", "Заказ"), их атрибуты (характеристики сущностей, например, "ФИО клиента", "Наименование товара") и связи между этими сущностями (например, "Клиент размещает Заказ"). ER-диаграммы в основном используется для проектирования базы данных, позволяя определить, какие таблицы будут созданы, какие в них будут поля и как они будут связаны друг с другом. Основные элементы диаграммы: сущности (обычно изображаются в виде прямоугольников), атрибуты (овалы, соединённые с сущностью), связи (ромбы, соединённые с участвующими сущностями). Также указываются кардинальности связей (один-к-одному, один-ко-многим, многие-ко-многим).

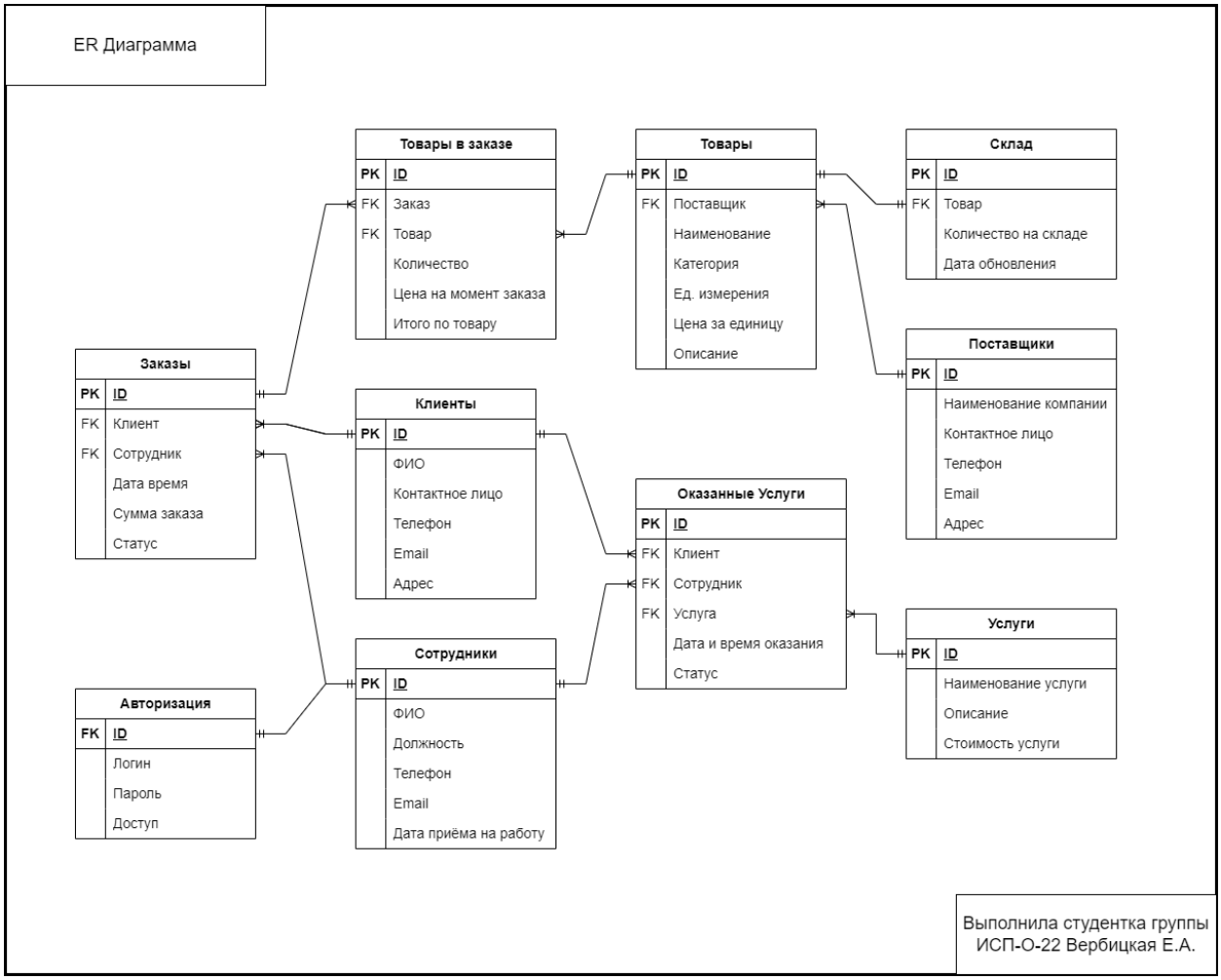

Рисунок 2.1. ER Диаграмма

На рисунке 2.1 представлена ER-диаграмма (Entity-Relationship Diagram), которая является графическим представлением структуры базы данных разрабатываемой информационной системы для ООО «Руссоль-центр». Диаграмма описывает сущности (таблицы), их атрибуты и связи между ними, что позволяет на этапе проектирования определить логическую структуру хранилища данных и обеспечить её целостность.

Сущности и их атрибуты:

- Товары: содержит информацию о номенклатуре товаров, предлагаемых компанией. Атрибуты включают уникальный идентификатор (ID), ссылку на поставщика (Поставщик — внешний ключ), Наименование, Категория, Ед. измерения, Цена за единицу и Описание.
- Поставщики: хранит данные о компаниях-поставщиках. Атрибуты: ID, Наименование компании, Контактное лицо, Телефон, Email, Адрес.

- Склад: отражает текущее состояние складских запасов. Содержит ID, ссылку на товар (Товар — внешний ключ), Количество на складе и Дата обновления.
- Услуги: описывает перечень услуг, которые может оказывать компания. Атрибуты: ID, Наименование услуги, Описание, Стоимость услуги.
- Клиенты: хранит информацию о клиентах компании. Атрибуты: ID, ФИО, Контактное лицо, Телефон, Email, Адрес.
- Сотрудники: содержит данные о персонале. Атрибуты: ID, ФИО, Должность, Телефон, Email, Дата приёма на работу.
- Авторизация: Таблица для управления доступом пользователей к системе. Содержит внешний ключ ID, который ссылается на таблицу Сотрудники, а также Логин, Пароль и Доступ (роль пользователя).
- Заказы: Основная таблица, фиксирующая заказы клиентов. Атрибуты: ID, ссылка на клиента (Клиент — внешний ключ), ссылка на сотрудника (Сотрудник — внешний ключ), Дата время, Сумма заказа, Статус.
- Товары в заказе: Связующая таблица, реализующая связь "многие ко многим" между Заказы и Товары. Содержит ID, ссылки на заказ (Заказ — внешний ключ) и товар (Товар — внешний ключ), а также Количество, Цена на момент заказа и итого по товару.
- Оказанные Услуги: Таблица, фиксирующая факт оказания услуг клиентам сотрудниками. Атрибуты: ID, ссылки на клиента (Клиент — внешний ключ), сотрудника (Сотрудник — внешний ключ) и услугу (Услуга — внешний ключ), Дата и время оказания, Статус.

Связи между сущностями:

- Между Товары и Поставщики установлена связь "один ко многим": один поставщик может поставлять множество товаров.
- Между Товары и Склад — связь "один к одному": каждый товар имеет одну запись о его количестве на складе.
- Между Товары и Товары в заказе — связь "один ко многим": один товар может входить в множество заказов.

- Между Заказы и Товары в заказе — связь "один ко многим": один заказ может содержать несколько позиций товаров.
- Между Заказы и Клиенты — связь "один ко многим": один клиент может оформить множество заказов.
- Между Заказы и Сотрудники — связь "один ко многим": один сотрудник может обслуживать множество заказов.
- Между Сотрудники и Авторизация — связь "один к одному": каждому сотруднику соответствует одна учётная запись для входа в систему.
- Между Клиенты и Оказанные Услуги — связь "один ко многим": один клиент может получать множество услуг.
- Между Сотрудники и Оказанные Услуги — связь "один ко многим": один сотрудник может оказывать множество услуг.
- Между Услуги и Оказанные Услуги — связь "один ко многим": одна услуга может быть оказана множеству клиентов.

ER-диаграмма служит основой для создания физической модели базы данных, гарантируя корректность структуры и логических связей между таблицами.

2.3.2. Контекстная диаграмма

Контекстная диаграмма (в нотации IDEF0 — диаграмма уровня A-0) — это верхнеуровневое представление системы, показывающее её как одну функцию и её взаимодействие с внешней средой. На диаграмме изображается главная функция системы (один прямоугольник), внешние сущности (например, "Клиент", "Поставщик", "Сотрудник"), и потоки данных, связывающие их с системой. Потоки данных делятся на:

- Входы (Input): Данные или ресурсы, поступающие в систему извне.
- Выходы (Output): Результаты работы системы, передаваемые во внешнюю среду.
- Управление (Control): Правила, регламенты, законы, влияющие на выполнение функции.

• Механизмы (Mechanism): Ресурсы, участвующие в выполнении функции (например, сотрудники, оборудование, ИТ-системы). Цель контекстной диаграммы — определить границы системы и понять, какие данные она получает и выдаёт, а также какие внешние факторы на неё влияют.

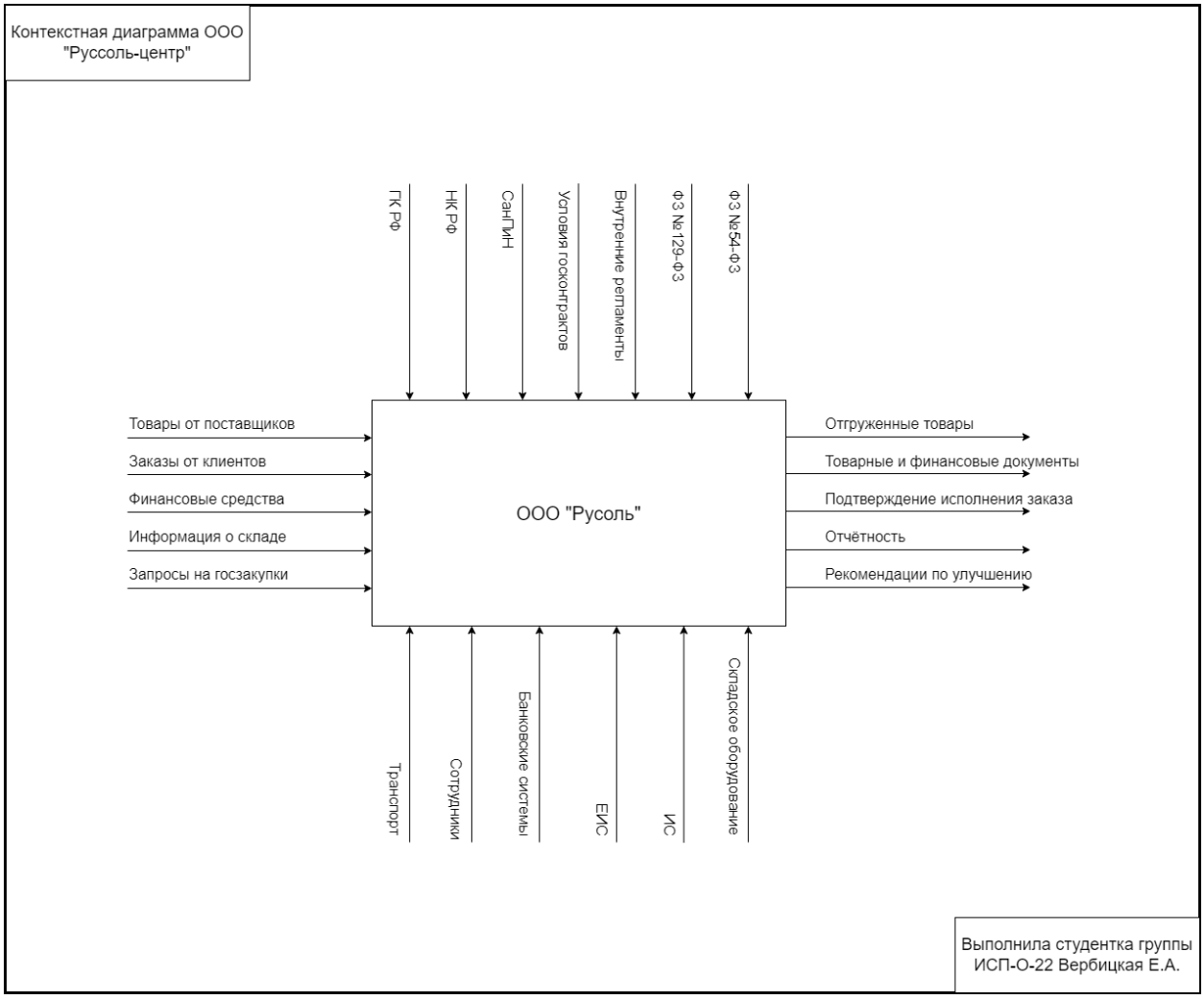

Рисунок 2.2. Контекстная диаграмма

Входы (Input):

• Товары от поставщиков: Физические товары, поступающие в систему от внешних поставщиков.

• Заказы от клиентов: Запросы на приобретение товаров, поступающие от клиентов.

• Финансовые средства: Денежные средства, поступающие от клиентов.

• Информация о складе: Данные о текущем состоянии остатков товаров на складе (возможно, из внешних источников или начального состояния).

- Запросы на госзакупки: Входящие данные о возможных государственных контрактах.

Выходы (Output):

- Отгруженные товары: Товары, физически покидающие систему и направляемые клиентам.
- Товарные и финансовые документы: Оформленные документы, связанные с продажами (накладные, счета и т.п.).
- Подтверждение исполнения заказа: Информация, передаваемая клиенту о выполнении его заказа.
- Отчётность: Документы, сформированные для внешних органов (налоговая, статистика и др.).
- Рекомендации по улучшению: Аналитические выводы, сгенерированные системой на основе накопленных данных.

Управление (Control):

- ГК РФ
- НК РФ
- СанПиН
- Условия госконтрактов
- Внутренние регламенты
- ФЗ №129-ФЗ
- ФЗ №54-ФЗ

Нормативно-правовые акты, внутренние регламенты и условия контрактов, которые регулируют и определяют правила функционирования системы, обязательства перед клиентами и поставщиками, а также требования к отчётности и кассовой дисциплине.

Механизмы (Mechanism):

- Транспорт: Средства доставки товаров.
- Сотрудники: Люди, участвующие в бизнес-процессах.

- Банковские системы: Внешние системы для обработки финансовых операций.
- ЕИС: Единая информационная система (например, Единая информационная система в сфере закупок).
- ИС: другие (внешние) информационные системы, с которыми взаимодействует система.
- Складское оборудование: Физическое оборудование, используемое на складе (весы, погрузчики и т.д.).

2.3.3. Диаграмма декомпозиции IDEF0

Диаграмма декомпозиции (в нотации IDEF0) — это способ детализации функций системы. Она представляет собой иерархию диаграмм, где сложная функция разбивается на более мелкие подфункции. Каждый блок на дочерней диаграмме также описывается с помощью четырёх компонентов: Вход (Input), Выход (Output), Управление (Control), Механизмы (Mechanism). Это позволяет пошагово, с возрастающей детализацией, описать, как система или процесс работает изнутри. Диаграммы связаны между собой, обеспечивая согласованность описания на разных уровнях.

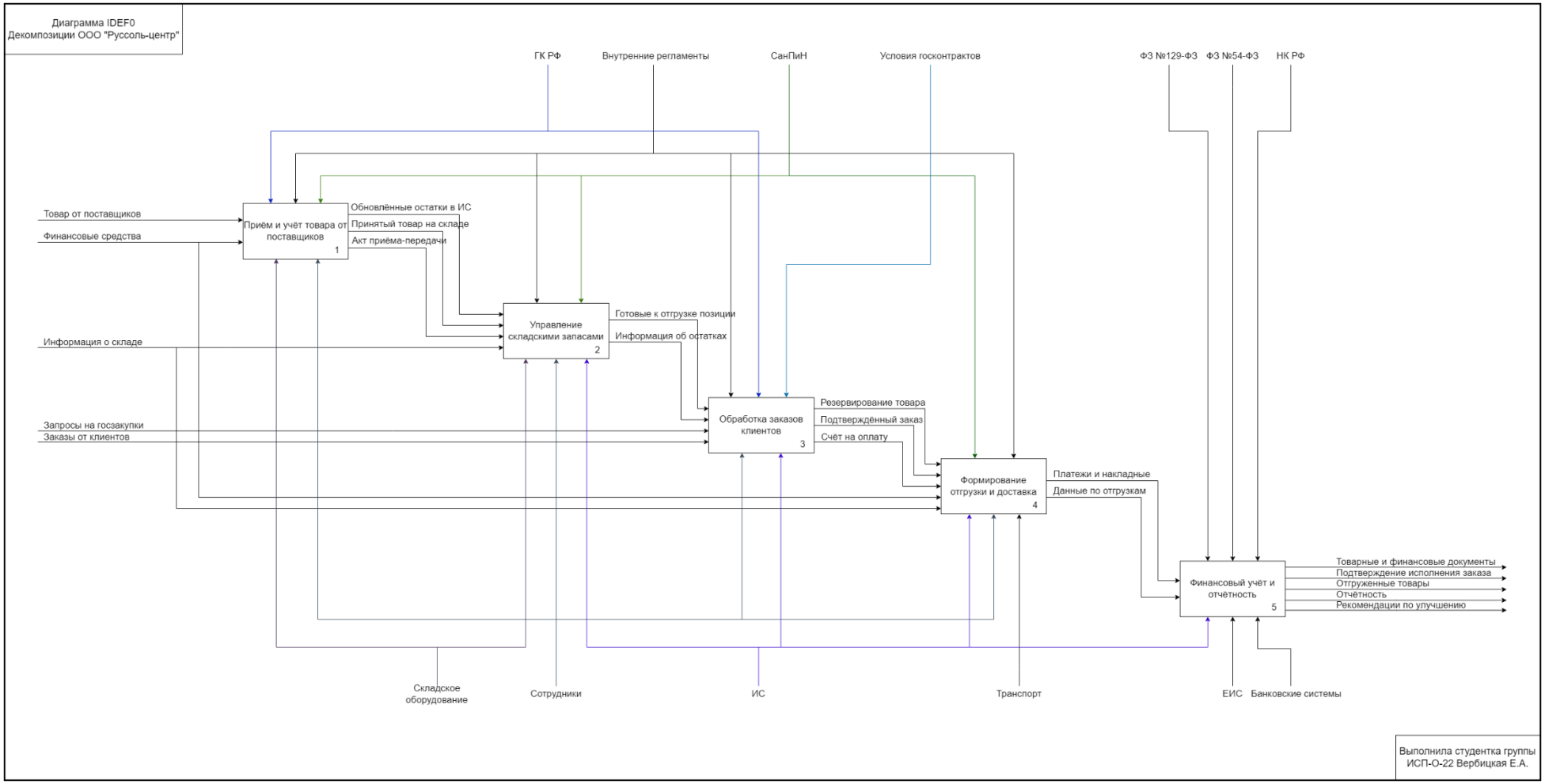

Рисунок 2.3. Диаграмма декомпозиции IDEF0

На рисунке 2.3 представлена диаграмма декомпозиции, выполненная в нотации IDEF0. Эта диаграмма является детализацией основной функции системы «ООО "Руссоль-центр»», описанной на контекстной диаграмме (см. рисунок 2.2). Она показывает, как единая функция системы разбивается на более мелкие подфункции, что позволяет глубже понять внутреннюю логику работы информационной системы и взаимодействие её компонентов.

Диаграмма состоит из четырёх основных блоков-процессов, каждый из которых описывается с помощью четырёх компонентов: Вход (Input), Выход (Output), Управление (Control) и Механизмы (Mechanism).

Блок 1: Обновление остатков в ИС

- Назначение: Этот процесс отвечает за актуализацию данных о наличии товаров на складе в информационной системе.
- Входы (Input): Товары от поставщиков, Запросы на госзакупки (как источник информации для планирования закупок).
- Выходы (Output): Информация об остатках (передаётся другим процессам).
- Управление (Control): ГК РФ, НК РФ, СанПиН, Условия госконтрактов, Внутренние регламенты.
- Механизмы (Mechanism): Складское оборудование, Сотрудники.

Блок 2: Управление складскими запасами

- Назначение: Основной процесс управления складом, включающий формирование заданий на сборку и контроль за движением товаров.
- Входы (Input): Информация об остатках (поступает из Блока 1), Заказы от клиентов.
- Выходы (Output): Информация о статусе заказа (передаётся в следующий блок), Формирование отгрузки и доставки (выходной поток).
- Управление (Control): ФЗ №129-ФЗ, ФЗ №54-ФЗ.
- Механизмы (Mechanism): Сотрудники, Транспорт.

Блок 3: Планирование товара

- Назначение: Процесс, связанный с анализом текущего состояния склада и планированием новых закупок.
- Входы (Input): Информация об остатках (из Блока 1), Заказы от клиентов.
- Выходы (Output): Плановые интервалы (дата по отгрузке) (информация для логистики и поставщиков).
- Управление (Control): Условия госконтрактов, Внутренние регламенты.
- Механизмы (Mechanism): Сотрудники, ИС (информационная система).

Блок 4: Финансовый учёт и отчётность

- Назначение: Финальный процесс, который фиксирует финансовые операции и генерирует необходимую отчётность.
- Входы (Input): Отгруженные товары (информация о реализованных товарах), Финансовые средства (поступления от клиентов).
- Выходы (Output): Товарные и финансовые документы, Подтверждение исполнения заказа, Отчётность, Рекомендации по улучшению.
- Управление (Control): ГК РФ, НК РФ, ФЗ №129-ФЗ, ФЗ №54-ФЗ.
- Механизмы (Mechanism): Банковские системы, ЕИС, ИС.

Взаимосвязи и потоки:

Стрелки на диаграмме показывают направление потоков данных, управления и механизмов между блоками и внешними сущностями. Например, Заказы от клиентов являются входом для Блока 2 и Блока 3, а Информация об остатках — это выход Блока 1 и вход для Блока 2 и Блока 3. Это демонстрирует, как данные передаются между различными частями системы, обеспечивая их согласованную работу. Диаграмма показывает, что все процессы тесно связаны и зависят друг от друга, а также от внешних факторов (нормативных актов и используемых ресурсов).

Диаграмма декомпозиции IDEF0 позволяет:

1) Детально проанализировать и спроектировать внутреннюю логику работы информационной системы.

2) Чётко определить границы и ответственность каждого процесса.

3) Увидеть, какие данные и ресурсы требуются для выполнения каждой функции.

4) Сформировать полное представление о том, как система будет работать на практике, что является основой для дальнейшей разработки модулей и программного кода.

2.3.4. Диаграмма потоков данных

DFD — это графическое представление потоков данных внутри системы. Диаграмма показывает, как данные перемещаются между различными процессами, хранилищами данных и внешними сущностями. Основные элементы:

- Внешние сущности (Terminator): Источники или получатели данных (например, "Клиент", "Поставщик").
- Процессы (Process): Действия, которые преобразуют входные данные в выходные (например, "Обработка заказа").
- Хранилища данных (Data Store): Места, где данные хранятся (например, "База данных клиентов").
- Потоки данных (Data Flow): Линии со стрелками, показывающие направление движения данных между элементами. DFD позволяет визуализировать логику обработки информации, не вдаваясь в детали реализации, и выявить все источники и потребителей данных.

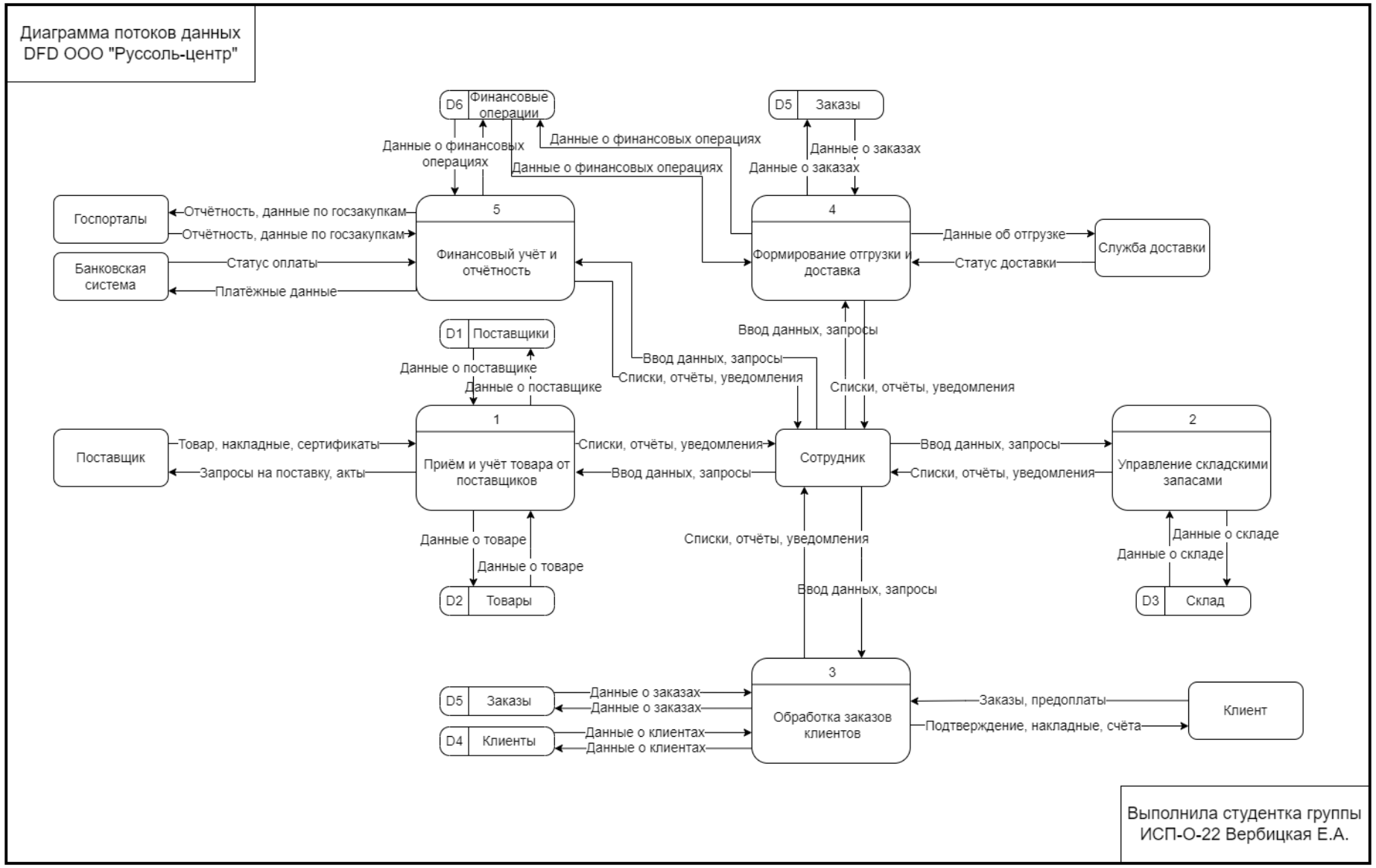

Рисунок 2.4. Диаграмма потоков данных

На рисунке 2.4 представлена диаграмма потоков данных (Data Flow Diagram, DFD) для информационной системы ООО «Руссоль-центр». Эта диаграмма является логической моделью, которая визуализирует, как данные перемещаются внутри системы между различными процессами, хранилищами данных и внешними сущностями. Она показывает, какие процессы выполняются над данными, какие источники и потребители этих данных существуют, а также где информация хранится. DFD помогает понять общую структуру обработки информации, не вдаваясь в детали реализации, что позволяет выявить ключевые точки взаимодействия и возможные узкие места.

Основные компоненты диаграммы:

Процессы (Processes): изображаются прямоугольниками с закругленными углами. Каждый процесс представляет собой действие или функцию, которая преобразует входные данные в выходные.

- Процесс 1: Приём и учёт товара от поставщиков. Отвечает за регистрацию товаров, поступающих от внешних поставщиков.
- Процесс 2: Управление складскими запасами. Обеспечивает учёт и управление остатками товаров на складе.
- Процесс 3: Обработка заказов клиентов. Основной процесс, который обрабатывает запросы клиентов, формирует заказы и взаимодействует с ними.
- Процесс 4: Формирование отгрузки и доставка. Отвечает за подготовку товаров к отправке клиентам и взаимодействие со службой доставки.
- Процесс 5: Финансовый учёт и отчётность. Обрабатывает все финансовые операции и генерирует необходимую отчётность.

Внешние сущности (External Entities): изображаются прямоугольниками. Это источники или получатели данных, которые находятся за пределами системы.

- Поставщик: Источник товаров и документов, поступающих в систему.

- Клиент: Получатель готовых товаров и документов, а также источник заказов.
- Сотрудник: Внутренний пользователь системы, который инициирует многие процессы (приём товаров, управление складом, обработка заказов, формирование отгрузки).
- Банковская система: Внешняя система, с которой взаимодействует система по части финансовых операций.
- Госпорталы: Внешние государственные системы, которым передаются отчётные данные (например, по госзакупкам).

Хранилища данных (Data Stores): изображаются прямоугольниками с открытым правым углом. Они представляют собой места, где данные хранятся в системе.

- D1 Поставщики: Хранилище информации о поставщиках.
- D2 Товары: Хранилище информации о номенклатуре товаров.
- D3 Склад: Хранилище данных об остатках товаров на складе.
- D4 Клиенты: Хранилище информации о клиентах.
- D5 Заказы: Хранилище всех заказов клиентов.
- D6 Финансовые операции: Хранилище данных о всех финансовых транзакциях.

Потоки данных (Data Flows): изображаются стрелками. Они показывают направление движения данных между компонентами. На стрелках указано название передаваемой информации.

Примеры: Запросы на поставку, Товар, накладные, сертификаты, Данные о товаре, Данные о складе, Заказы, предоплаты, Подтверждение, накладные, счёт, Статус доставки, Отчётность, Платёжные данные.

Логика работы системы:

Диаграмма показывает, что основным "двигателем" системы являются Сотрудники, которые взаимодействуют со всеми основными процессами. Процесс начинается с Поставщика, который предоставляет товары и

документы, которые через процесс 1 попадают в хранилище D2 Товары и D3 Склад. Информация о поставщике хранится в D1 Поставщики.

Процесс 3 (Обработка заказов клиентов) получает данные от Клиента (заказы) и из хранилищ D4 Клиенты и D5 Заказы. После обработки заказа он передаёт информацию в процесс 4 (Формирование отгрузки и доставка), который взаимодействует со Службой доставки и обновляет данные в хранилище D5 Заказы. Данные о заказах и отгрузке также передаются в процесс 5 (Финансовый учёт и отчётность).

Процесс 5 получает данные о финансовых операциях от Банковской системы и от других процессов, а также от хранилища D6 Финансовые операции. Он формирует отчёты, которые передаются в Госпорталы и другим внешним сущностям.

Таким образом, DFD демонстрирует полный цикл работы системы: от получения товара от поставщика до отгрузки клиенту и формирования финансовой отчётности, подчеркивая роль сотрудника как центрального актора, управляющего всеми ключевыми процессами.

2.3.5. Диаграмма вариантов использования (Use Case Diagram)

Use Case диаграмма — это элемент Unified Modeling Language (UML), описывающий функции системы с точки зрения взаимодействия с акторами. Актор — это внешняя сущность (например, человек, другая система), которая взаимодействует с системой. Use Case (вариант использования) — это конкретная цель, которую актор преследует при взаимодействии с системой (например, "Создать заказ", "Просмотреть отчёт"). Диаграмма показывает, *какие* функции предоставляет система и *кто* их использует. Основные элементы:

- Акторы (Actor): изображаются в виде стилизованных человечков.
- Use Case (Вариант использования): изображаются в виде эллипсов.
- Система (System Boundary): обычно прямоугольник, ограничивающий варианты использования.

- Отношения: Линии, соединяющие акторов и use case, показывающие взаимодействие. Также могут быть отношения include и extend между самими use case. Цель диаграммы — определить функциональные требования к системе с позиции пользователя.

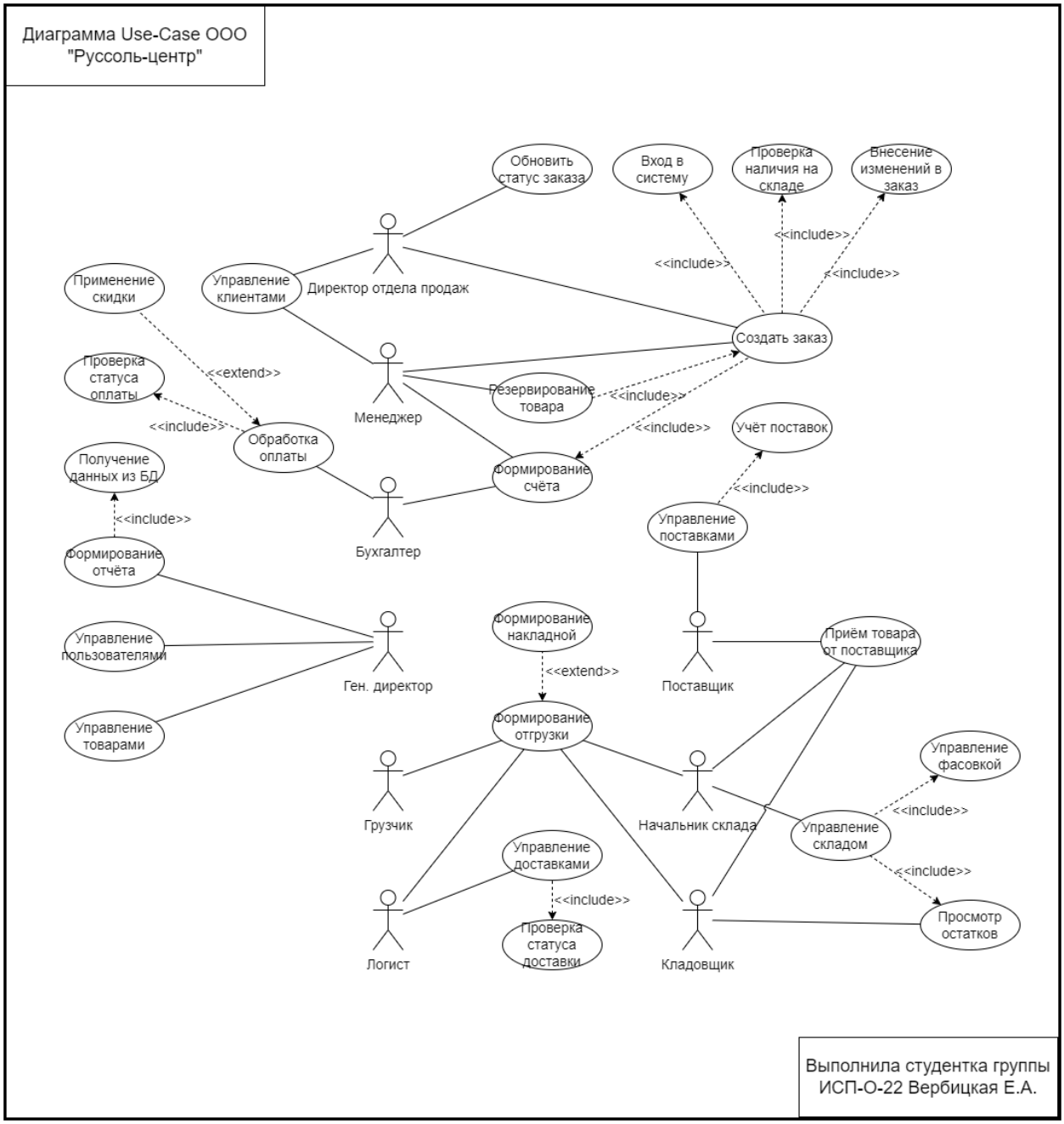

Рисунок 2.5. Диаграмма вариантов использования

На рисунке 2.5 представлена диаграмма вариантов использования (Use Case Diagram) для информационной системы ООО «Руссоль-центр». Эта диаграмма, выполненная в нотации UML (Unified Modeling Language), является одним из ключевых инструментов на этапе проектирования,

позволяющим визуализировать функциональные требования к системе с точки зрения её пользователей. Она показывает, кто (акторы) взаимодействует с системой и что (варианты использования) он может сделать, не углубляясь в детали реализации. Это помогает сформировать общее понимание того, какие функции должна предоставлять система для каждой роли пользователя.

Основные элементы диаграммы:

Акторы (Actors): изображаются в виде стилизованных человечков и представляют собой внешние сущности (пользователей или другие системы), которые взаимодействуют с разрабатываемой ИС. На данной диаграмме выделены следующие акторы:

- Директор отдела продаж: отвечает за управление клиентами и создание заказов.
- Менеджер: Основной пользователь, работающий с заказами, формирующий счета и управляющий поставками.
- Бухгалтер: обрабатывает оплаты и формирует отчёты.
- Ген. директор: имеет полный доступ к системе, включая управление пользователями и товарами.
- Поставщик: взаимодействует с системой для передачи информации о поступлении товара.
- Начальник склада: управляет складом и фасовкой.
- Кладовщик: проводит операции по приёму и отпуску товаров.
- Логист: участвует в формировании отгрузки.
- Грузчик: выполняет физические операции по погрузке/разгрузке.

Варианты использования (Use Cases): изображаются в виде эллипсов и описывают конкретную цель, которую актор преследует при взаимодействии с системой. Каждый use case представляет собой одну законченную функцию.

Примеры: создать заказ, обновить статус заказа, Формирование счёта, Управление клиентами, Приём товара от поставщика, Управление складом, Просмотр остатков и др.

Отношения между элементами:

- Ассоциация (Association): Прямая линия, соединяющая актора с use case, показывает, что актор использует данный вариант использования.
- Отношение включения (<<include>>): Пунктирная стрелка с надписью <<include>> указывает, что один use case (базовый) всегда включает в себя другой (включаемый). Например, создать заказ включает в себя Проверка наличия на складе, так как без проверки наличие товара невозможно оформить заказ.
- Отношение расширения (<<extend>>): Пунктирная стрелка с надписью <<extend>> указывает, что один use case (расширяющий) может дополнить другой (базовый) в определённых условиях. Например, Обработка оплаты может быть расширена действием Применение скидки, если это предусмотрено бизнес-логикой.

Анализ диаграммы:

Диаграмма демонстрирует, что система имеет сложную структуру с множеством ролей и взаимосвязанных функций. Можно выделить несколько ключевых зон ответственности:

1) Зона управления продажами и клиентами: Акторы Директор отдела продаж и Менеджер взаимодействуют с системой для создания и управления заказами, а также для работы с клиентской базой.
2) Зона финансового учёта: Актор Бухгалтер работает с оплатами и отчётностью.
3) Зона складского учёта и логистики: Акторы Начальник склада, Кладовщик, Логист, Грузчик и Поставщик обеспечивают цикл движения товара — от его приёма до отгрузки.
4) Зона администрирования: Актор Ген. директор имеет максимальный доступ ко всем функциям, включая управление пользователями и справочниками товаров.

Диаграмма Use Case служит основой для дальнейшего проектирования. Она позволяет:

1) Чётко определить функциональные требования к системе с позиции пользователя.

2) Установить границы системы и понять, кто и как будет ею пользоваться.

3) Сформировать список тестовых сценариев для каждого варианта использования.

4) Обеспечить единое понимание между разработчиками, заказчиком и другими заинтересованными сторонами относительно того, какие функции должны быть реализованы.

Таким образом, данная диаграмма предоставляет комплексное представление о функциональности информационной системы ООО «Руссоль-центр», отражая все ключевые процессы и роли пользователей, что является необходимым шагом для успешной разработки программного обеспечения.

2.4. Макеты интерфейса приложения

Макеты интерфейса приложения (также известные как mockups или wireframes) представляют собой визуальные схемы, которые демонстрируют структуру, расположение элементов и логику взаимодействия пользователя с системой. Они создаются на этапе проектирования и служат важным инструментом для:

- Визуализации будущего пользовательского интерфейса (UI).
- Согласования функциональности и внешнего вида с заинтересованными сторонами (заказчиком, разработчиками).
- Выявления потенциальных проблем с юзабилити до начала программирования.
- Формирования четких требований к разработке интерфейса.

В рамках проекта разработки информационной системы для ООО «Руссоль-центр» макеты интерфейса были созданы для ключевых функций, определённых в требованиях. Макеты отражают основные сценарии

использования, предусмотренные для различных ролей пользователей: администратор, менеджер по заказам, менеджер по услугам.

При создании макетов учитывались следующие принципы:

- Интуитивность: Интерфейс должен быть понятным и не требовать длительного обучения. Навигация и расположение элементов должны быть логичными.
- Единообразие: Схожие элементы (кнопки, поля ввода, таблицы) должны иметь единый стиль и поведение.
- Доступность: Макеты ориентированы на пользователей, работающих в Windows, с использованием привычных элементов управления.
- Фокус на задачу: Интерфейс каждого окна должен быть сосредоточен на выполнении конкретной задачи, избегая излишней информации и элементов.

Окно авторизации (рис. 2.6) является стартовым экраном приложения. Его основная функция — аутентификация пользователя, то есть проверка его личности на основе введённых учётных данных (логина и пароля). Только после успешной авторизации пользователь получает доступ к функционалу системы, соответствующему его роли.

Интерфейс: Окно содержит поля для ввода Логина и Пароля, а также кнопку Войти для подтверждения введённых данных.

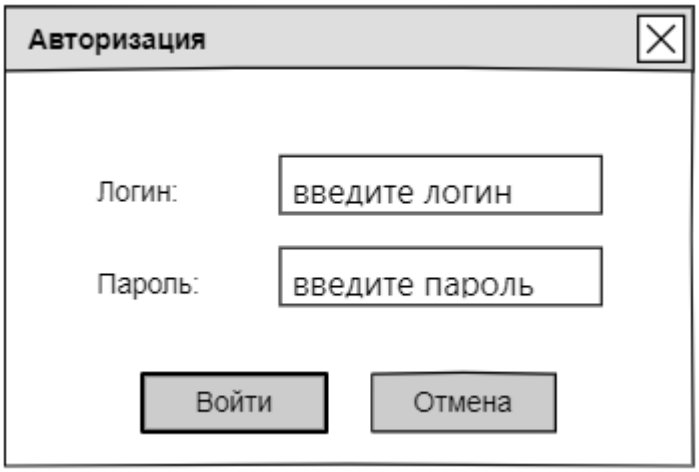

Рисунок 2.6. Макет окна авторизации

Главное окно (рис. 2.7) является центральной точкой входа в информационную систему после успешной авторизации пользователя. Оно открывается автоматически и служит навигационным хабом, откуда пользователь может перейти к выполнению основных задач, доступных для его роли. Это окно обеспечивает единый и понятный интерфейс для всех категорий пользователей: администратора, менеджера по заказам и менеджера по услугам.

Интерфейс: В верхней части окна отображается приветственное сообщение с именем текущего пользователя (например, "Добро пожаловать, Козлов Артем Михайлович"), что создаёт персонализированное ощущение и подтверждает успешность входа. Основная часть окна содержит сетку из кнопок-контроллеров, каждая из которых ведёт к определённому функциональному модулю системы. Названия кнопок соответствуют ключевым разделам: Заказы, Услуги, Склад, Оказанные услуги, Поставщики, Сотрудники, Авторизация, Клиенты, Товары.

Адаптация под роли:

- Администратор: для администратора главное окно является полным и предоставляет доступ ко всем девяти кнопкам. Это позволяет ему управлять всеми аспектами системы: от создания новых заказов и услуг до управления справочниками (клиенты, сотрудники, товары, поставщики), настройки системы (Авторизация) и контроля за складскими запасами.
- Менеджер по заказам: для этой роли главное окно адаптируется и показывает только те кнопки, которые относятся к его зоне ответственности. На рисунке 2.8 видно, что доступны кнопки Заказы, Поставщики, Склад, Клиенты и Товары. Это даёт менеджеру необходимый функционал для работы с клиентами, оформления заказов, проверки наличия товаров на складе и взаимодействия с поставщиками, исключая функции, не относящиеся к его обязанностям (например, управление сотрудниками или оказанными услугами).

• Менеджер по услугам: для этого пользователя главное окно также адаптируется, предоставляя доступ только к тем функциям, которые ему необходимы. На рисунке 2.9 показано, что доступны кнопки Оказанные услуги, Услуги, Сотрудники и Клиенты. Это позволяет менеджеру по услугам фокусироваться на своём основном направлении — учёте и оформлении услуг, взаимодействии с клиентами и сотрудниками, не отвлекаясь на другие модули.

Таким образом, главное окно реализует принцип разграничения прав доступа, динамически изменяя свой интерфейс в зависимости от роли вошедшего пользователя. Это обеспечивает удобство использования, так как каждый пользователь видит только те функции, которыми он должен пользоваться, что снижает риск ошибок и упрощает навигацию.

Рисунок 2.7. Макет главного окна - администратора

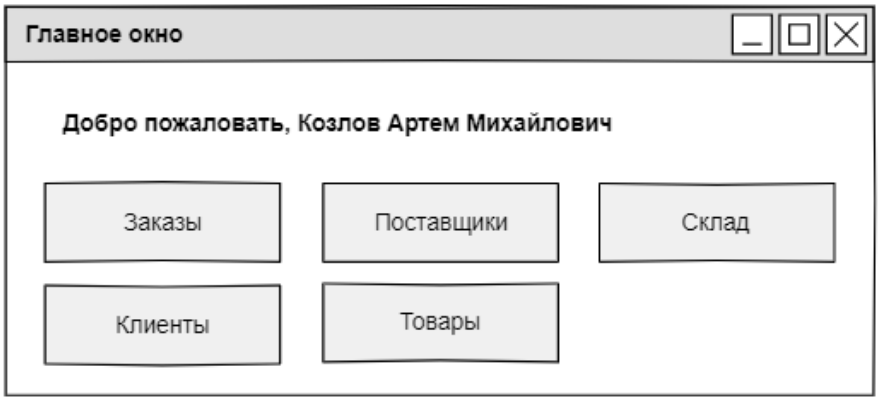

Рисунок 2.8. Макет главного окна – менеджера по заказам

Рисунок 2.9. Макет главного окна – менеджера по услугам

На рисунке 2.10 представлен макет окна «Заказы», которое является основным рабочим экраном для менеджера по заказам и администратора. Его цель — предоставить пользователю общий обзор всех созданных заказов, а также инструменты для их управления.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Заказы» и кнопками управления (свернуть, развернуть, закрыть). В верхней части окна расположен блок поиска: поле ввода с меткой Поиск: позволяет быстро фильтровать список заказов по любому из столбцов (например, по ID, имени клиента или статусу).

Основная часть: занимает большую часть окна и представляет собой таблицу со следующими столбцами:

- ID — уникальный номер заказа.
- Клиент — наименование или ФИО клиента, оформившего заказ.
- Сотрудник — имя сотрудника, который работал с этим заказом.
- Дата заказа — дата и время создания заказа.
- Сумма — общая стоимость заказа.
- Статус — текущее состояние заказа (Новый, Подтверждён, Отгружен и т.д.).
- Товары — краткий перечень товаров, входящих в заказ (например, "Соль, Сахар").

Ячейки таблицы заполняются данными из базы данных. Пользователь может просматривать данные, сортировать строки по любому столбцу (щелчком по заголовку) и выбирать строку для дальнейших действий.

Панель управления: В нижней части окна расположена панель с четырьмя основными кнопками:

- Добавить — открывает окно для создания нового заказа.
- Изменить — открывает окно редактирования выбранного заказа.
- Удалить — удаляет выбранный заказ (после подтверждения).
- Обновить — обновляет данные в таблице, отображая последние изменения в базе данных.

Этот макет обеспечивает удобное и эффективное управление списком заказов, позволяя пользователю быстро находить нужные записи, просматривать их детали и выполнять основные операции (добавление, изменение, удаление) с помощью интуитивно понятных кнопок.

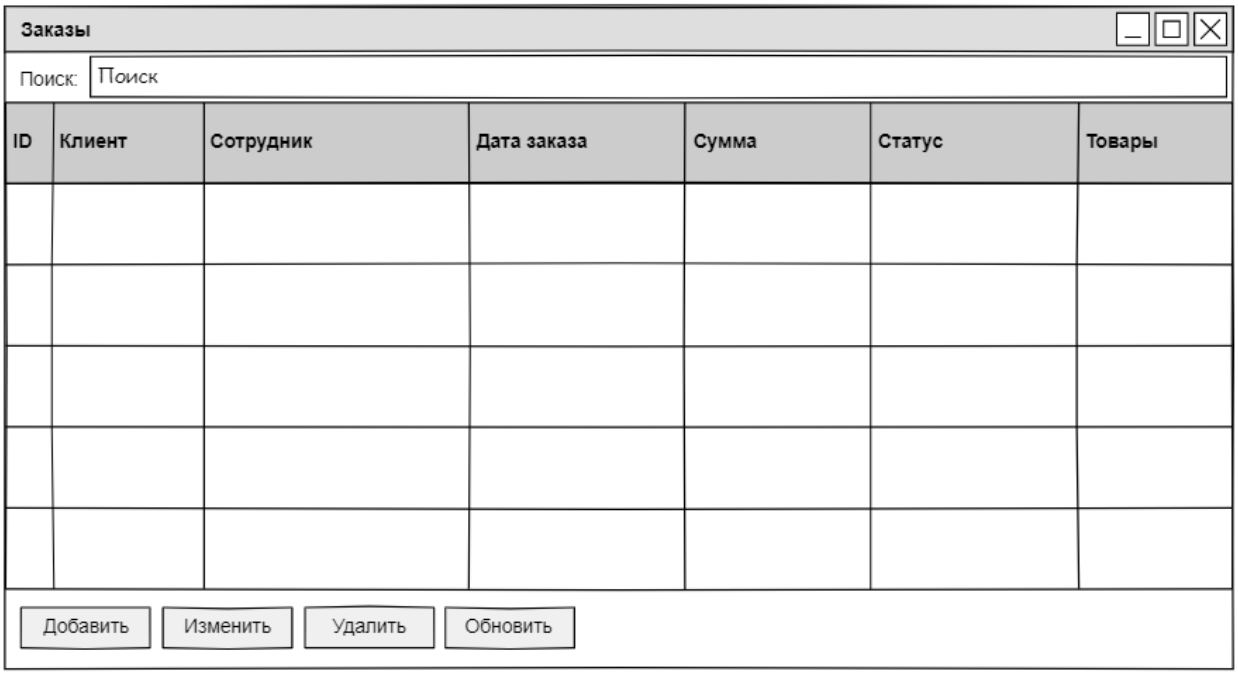

Рисунок 2.10. Макет окна Заказы

На рисунке 2.11 представлен макет окна «Заказы», которое используется как для создания нового заказа, так и для редактирования существующего. Это окно является ключевым рабочим инструментом для менеджера по заказам и администратора.

Интерфейс: Окно содержит поля для ввода и отображения основных данных о заказе:

- Клиент: Поле с выпадающим списком (ComboBox) для выбора клиента из справочника.
- Сотрудник: Поле с выпадающим списком для выбора сотрудника, ответственного за заказ.
- Дата заказа: Поле с календарём (DateTimePicker), позволяющее выбрать дату и время оформления заказа. В новом заказе здесь будет текущая дата, при редактировании — дата оригинального заказа.
- Сумма заказа: Текстовое поле, которое обычно рассчитывается автоматически на основе стоимости товаров и их количества, но может быть отредактировано вручную (например, для применения скидки).
- Статус: Текстовое поле или выпадающий список, отображающий текущий статус заказа (по умолчанию "Новый").
- Блок "Товары": ниже расположена таблица, в которой перечисляются товары, входящие в заказ. Столбцы таблицы: Наименование товара и Количество. Пользователь может добавлять новые строки в эту таблицу, выбирая товар и указывая количество. При редактировании существующего заказа, все уже добавленные товары подгружаются из базы данных в эту таблицу.

Панель управления: В нижней части окна находится кнопка Добавить, которая выполняет две функции:

- При создании нового заказа — сохраняет его в базе данных.
- При редактировании существующего заказа — обновляет все данные (информацию о клиенте, сотруднике, статусе, сумму) и состав товаров в соответствии с внесёнными изменениями. После нажатия кнопки окно закрывается, а в списке заказов отображаются обновлённые данные.

Этот макет обеспечивает удобный и логичный интерфейс для работы с заказами, позволяя пользователю легко заполнять необходимую информацию и управлять составом заказа.

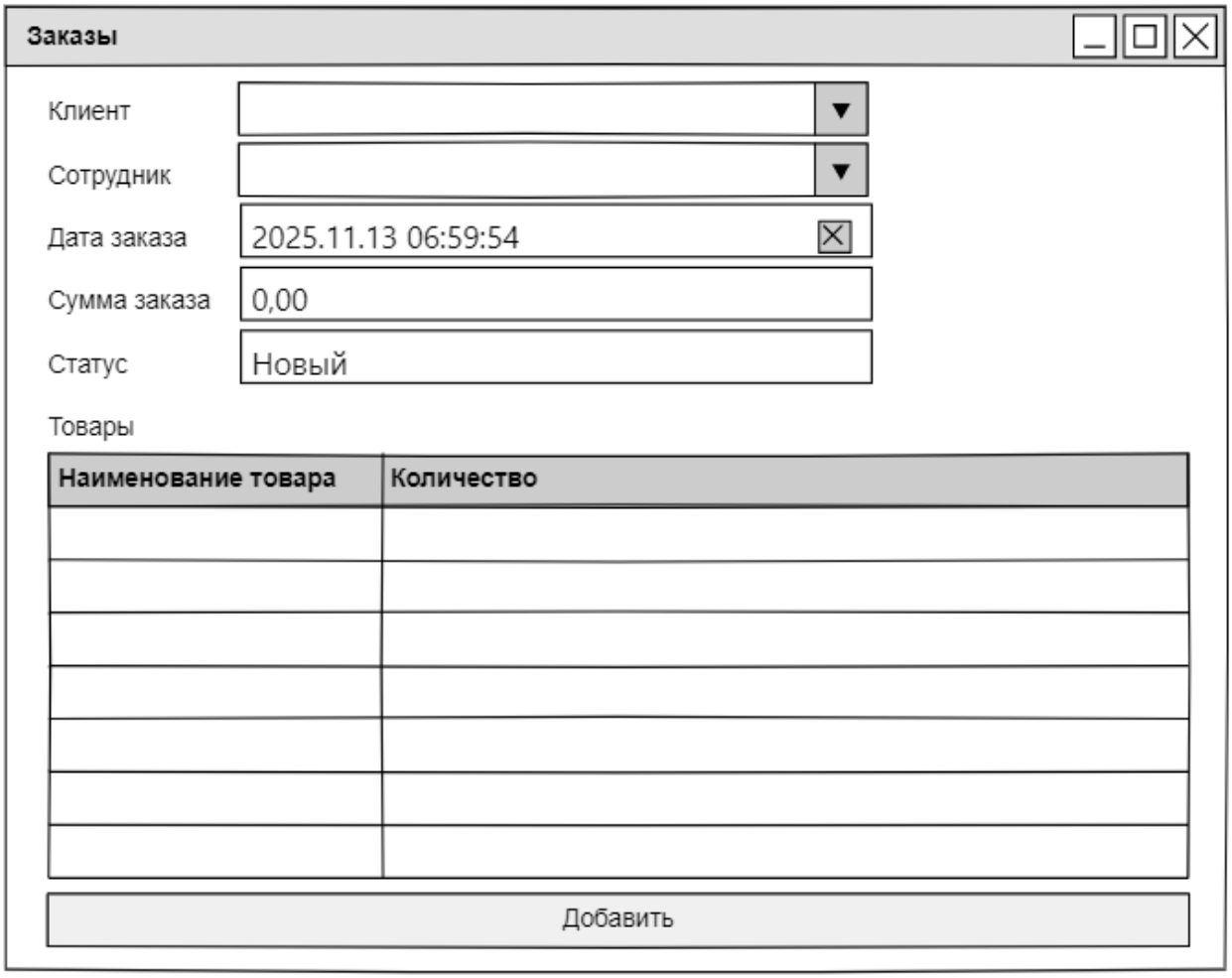

Рисунок 2.11. Макет окна создания/редактирования заказа

На рисунке 2.12 представлен макет окна «Услуги», которое является основным рабочим экраном для администратора и менеджера по услугам. Его цель — предоставить пользователю общий обзор всех услуг, доступных в системе, а также инструменты для их управления.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Услуги» и кнопками управления (свернуть, развернуть, закрыть). В верхней части окна расположен блок поиска: поле ввода с меткой Поиск: позволяет быстро фильтровать список услуг по любому из столбцов (например, по ID или наименованию).

Основная часть: занимает большую часть окна и представляет собой таблицу со следующими столбцами:

- ID — уникальный номер услуги.
- Наименование — название услуги.

- Описание — подробное описание услуги.
- Цена — стоимость услуги.

Ячейки таблицы заполняются данными из базы данных. Пользователь может просматривать данные, сортировать строки по любому столбцу (щелчком по заголовку) и выбирать строку для дальнейших действий.

Панель управления: В нижней части окна расположена панель с четырьмя основными кнопками:

- Добавить — открывает окно для создания новой услуги.
- Изменить — открывает окно редактирования выбранной услуги.
- Удалить — удаляет выбранную услугу (после подтверждения).
- Обновить — обновляет данные в таблице, отображая последние изменения в базе данных.

Этот макет обеспечивает удобное и эффективное управление справочником услуг, позволяя пользователю быстро находить нужные записи, просматривать их детали и выполнять основные операции (добавление, изменение, удаление) с помощью интуитивно понятных кнопок.

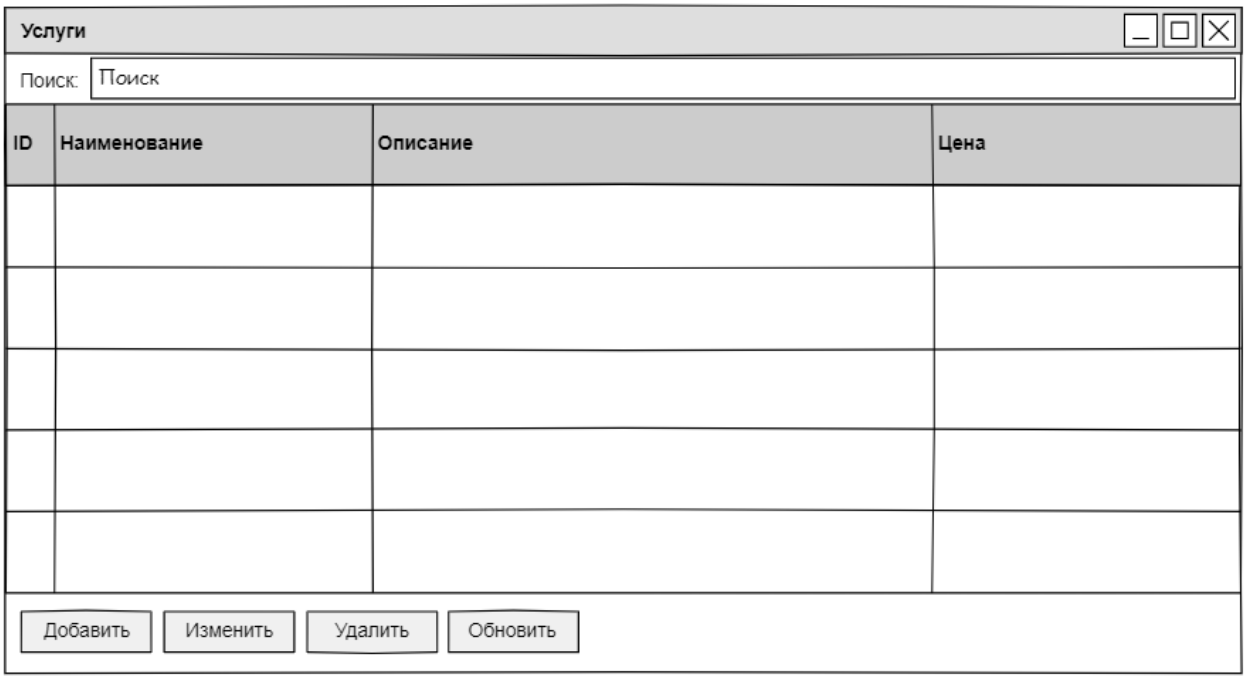

Рисунок 2.12. Макет окна услуги

На рисунке 2.13 представлен макет окна, предназначенного для создания новой услуги или редактирования существующей в информационной системе

ООО «Руссоль-центр». Это окно является ключевым для администратора и менеджера по услугам, позволяя им управлять справочником услуг.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Услуги» и кнопкой закрытия. Внутри расположены три поля ввода:

- Наименование: Текстовое поле для ввода названия услуги (например, "Консультация по подбору соли").
- Описание: Текстовое поле, в котором можно указать подробное описание услуги, её особенности или условия оказания.
- Цена: Текстовое поле для указания стоимости услуги. Обычно здесь вводится числовое значение.

Логика работы:

- При создании новой услуги пользователь заполняет все поля и нажимает кнопку Добавить. После этого данные сохраняются в базе данных, и окно закрывается.
- При редактировании существующей услуги окно открывается с уже подгруженными данными из базы данных. Пользователь может изменить любое поле (наименование, описание, цену) и снова нажать добавить (в данном контексте это означает "Сохранить изменения"). Система обновляет запись в базе данных.
- Кнопка Добавить является универсальной: она используется как для создания новой записи, так и для сохранения изменений в существующей.

Этот макет обеспечивает простой и понятный интерфейс для управления справочником услуг, что является важным элементом для расширения функционала системы и предоставления клиентам дополнительных возможностей.

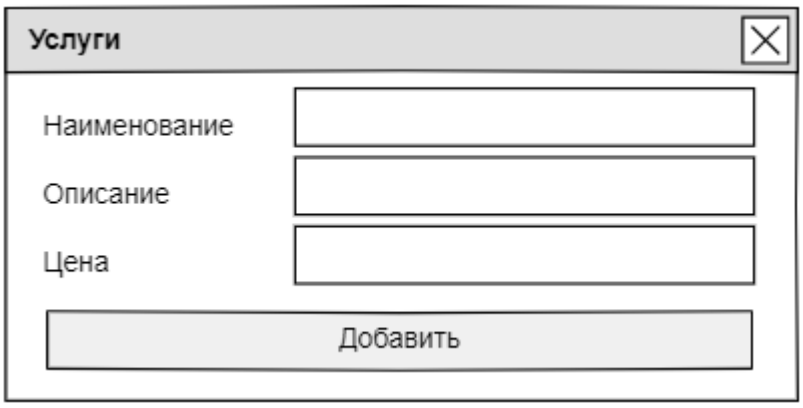

Рисунок 2.13. Макет окна создания/редактирования услуги

На рисунке 2.14 представлен макет окна «Склад», которое является основным рабочим экраном для кладовщика, начальника склада и администратора. Его цель — предоставить пользователю общий обзор текущего состояния запасов на складе, а также инструменты для их управления.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Склад» и кнопками управления (свернуть, развернуть, закрыть). В верхней части окна расположен блок поиска: поле ввода с меткой Поиск: позволяет быстро фильтровать список товаров по любому из столбцов (например, по ID или наименованию товара).

Основная часть: занимает большую часть окна и представляет собой таблицу со следующими столбцами:

- ID — уникальный номер записи о товаре на складе.
- Товар — наименование товара, ссылка на справочник товаров.
- Количество на складе — актуальное количество единиц данного товара, находящееся на складе.
- Дата изменения — дата и время последнего изменения количества (поступления или отпуска).

Ячейки таблицы заполняются данными из базы данных. Пользователь может просматривать данные, сортировать строки по любому столбцу (щелчком по заголовку) и выбирать строку для дальнейших действий.

Панель управления: В нижней части окна расположена панель с четырьмя основными кнопками:

- Добавить — открывает окно для добавления новой записи о товаре на склад (например, при поступлении нового товара или создании первой записи).
- Изменить — открывает окно редактирования выбранной записи, позволяя изменить количество товара на складе (например, при проведении инвентаризации или корректировке ошибки).
- Удалить — удаляет выбранную запись о товаре (после подтверждения). Это действие должно использоваться с осторожностью.
- Обновить — обновляет данные в таблице, отображая последние изменения в базе данных.

Этот макет обеспечивает удобное и эффективное управление складскими запасами, позволяя пользователю быстро находить нужные товары, просматривать их остатки и выполнять основные операции (добавление, изменение, удаление) с помощью интуитивно понятных кнопок.

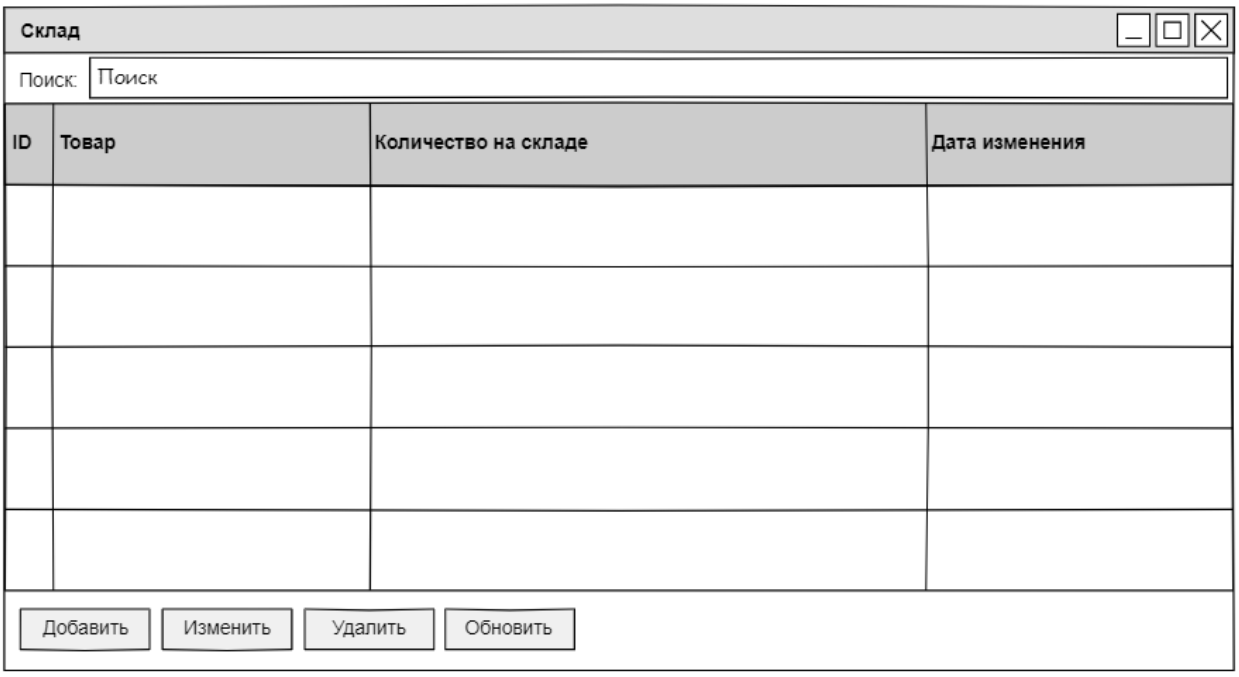

Рисунок 2.14. Макет окна склад

На рисунке 2.15 представлен макет окна, предназначенного для создания новой записи о товаре на складе или редактирования существующей в

информационной системе ООО «Руссоль-центр». Это окно является ключевым для кладовщика и администратора, позволяя им управлять данными о наличии товаров.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Склад» и кнопкой закрытия. Внутри расположены два поля:

- Товар: Поле с выпадающим списком (ComboBox), которое позволяет выбрать товар из справочника.
- Количество на складе: Текстовое поле, в котором пользователь вводит текущее количество единиц выбранного товара, находящегося на складе.

Логика работы:

- При создании новой записи пользователь выбирает товар и указывает его количество на складе, затем нажимает кнопку Добавить. После этого данные сохраняются в базе данных, и окно закрывается.
- При редактировании существующей записи окно открывается с уже подгруженными данными (выбранным товаром и его текущим количеством). Пользователь может изменить количество и снова нажать добавить (в данном контексте это означает "Сохранить изменения"). Система обновляет запись в базе данных.
- Кнопка Добавить является универсальной: она используется как для создания новой записи, так и для сохранения изменений в существующей.

Этот макет обеспечивает простой и понятный интерфейс для управления складскими запасами, что является важным элементом для точного учёта и планирования закупок.

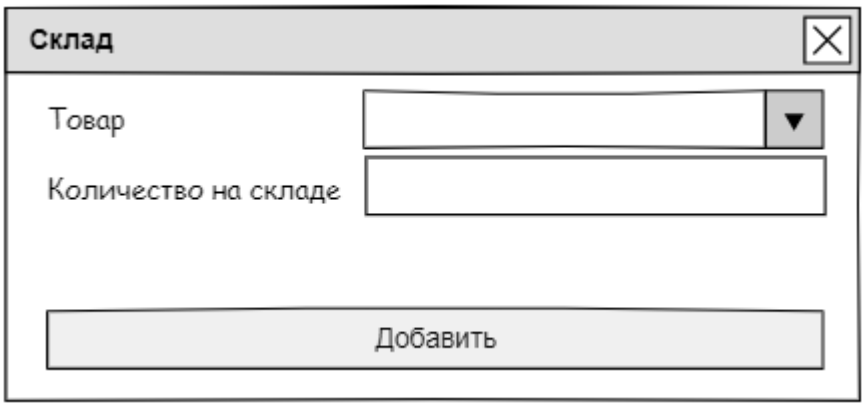

Рисунок 2.15. Макет окна создания/редактирования товара на складе

На рисунке 2.16 представлен макет окна «Оказанные услуги», которое является основным рабочим экраном для менеджера по услугам и администратора. Его цель — предоставить пользователю общий обзор всех услуг, которые были оказаны клиентам, а также инструменты для их управления.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Оказанные услуги» и кнопками управления (свернуть, развернуть, закрыть). В верхней части окна расположен блок поиска: поле ввода с меткой Поиск: позволяет быстро фильтровать список услуг по любому из столбцов (например, по ID, имени клиента или наименованию услуги).

Основная часть: занимает большую часть окна и представляет собой таблицу со следующими столбцами:

- ID — уникальный номер записи об оказанной услуге.
- Клиент — наименование или ФИО клиента, которому была оказана услуга.
- Сотрудник — имя сотрудника, который оказал услугу.
- Услуга — наименование оказанной услуги.
- Дата оказания — дата и время, когда услуга была оказана.
- Статус — текущее состояние записи (Новая, Выполнена, Отменена и т.д.).

Ячейки таблицы заполняются данными из базы данных. Пользователь может просматривать данные, сортировать строки по любому столбцу (щелчком по заголовку) и выбирать строку для дальнейших действий.

Панель управления: В нижней части окна расположена панель с четырьмя основными кнопками:

- Добавить — открывает окно для создания новой записи об оказанной услуге.
- Изменить — открывает окно редактирования выбранной записи.
- Удалить — удаляет выбранную запись (после подтверждения).
- Обновить — обновляет данные в таблице, отображая последние изменения в базе данных.

Этот макет обеспечивает удобное и эффективное управление списком оказанных услуг, позволяя пользователю быстро находить нужные записи, просматривать их детали и выполнять основные операции (добавление, изменение, удаление) с помощью интуитивно понятных кнопок.

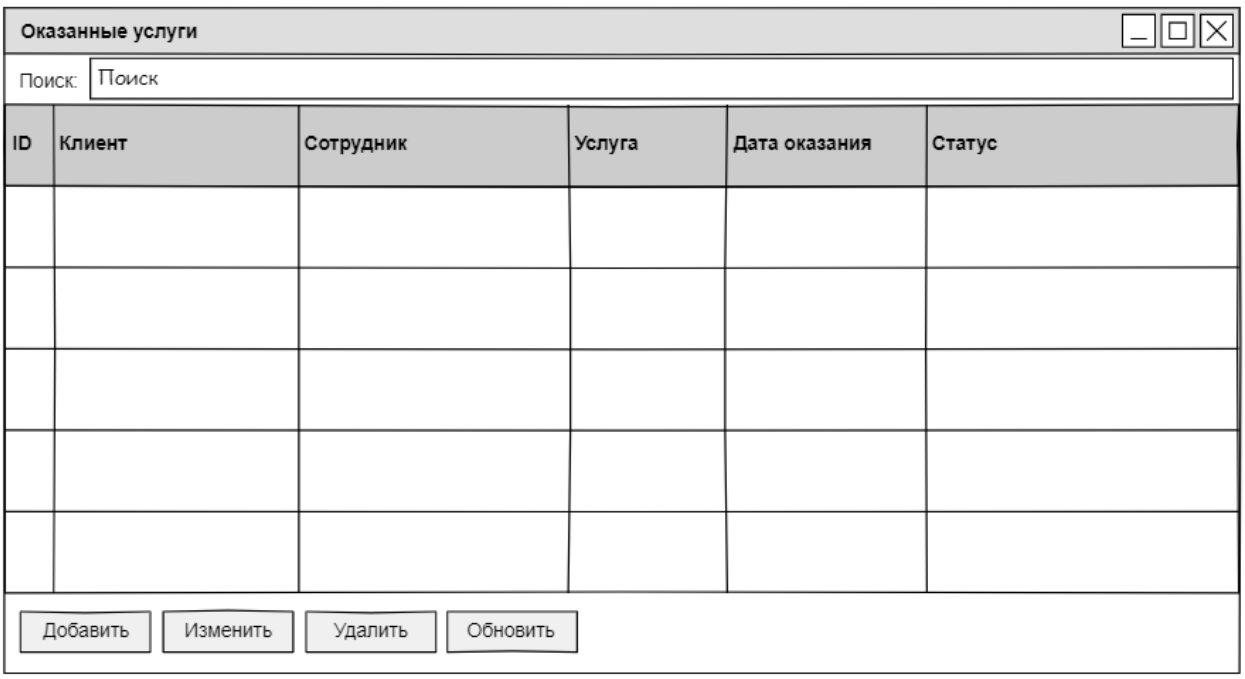

Рисунок 2.16. Макет окна оказанные услуги

На рисунке 2.17 представлен макет окна, предназначенного для создания новой записи об оказанной услуге или редактирования существующей в информационной системе ООО «Руссоль-центр». Это окно является

ключевым для менеджера по услугам и администратора, позволяя им фиксировать факт выполнения работы.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Оказанные услуги» и кнопкой закрытия. Внутри расположены пять полей:

- Клиент: Поле с выпадающим списком (ComboBox), которое позволяет выбрать клиента из справочника.
- Сотрудник: Поле с выпадающим списком, позволяющее выбрать сотрудника, который оказал услугу.
- Услуга: Поле с выпадающим списком, в котором выбирается наименование самой услуги из соответствующего справочника.
- Дата оказания: Текстовое поле, в котором пользователь может указать дату и время, когда услуга была оказана клиенту.
- Статус: Текстовое поле, отображающее текущий статус записи (например, "Новая", "Выполнена", "Отменена"). При создании новой услуги статус устанавливается по умолчанию.

Логика работы:

- При создании новой записи пользователь последовательно заполняет все поля (выбирает клиента, сотрудника, услугу, указывает дату) и нажимает кнопку Добавить. После этого данные сохраняются в базе данных, и окно закрывается.
- При редактировании существующей записи окно открывается с уже подгруженными данными из базы данных. Пользователь может изменить любое поле (например, исправить дату или сменить статус) и снова нажать добавить (в данном контексте это означает "Сохранить изменения"). Система обновляет запись в базе данных.
- Кнопка Добавить является универсальной: она используется как для создания новой записи, так и для сохранения изменений в существующей.

Этот макет обеспечивает простой и понятный интерфейс для управления записями об оказанных услугах, что является важным элементом для точного учёта деятельности компании и формирования отчётности.

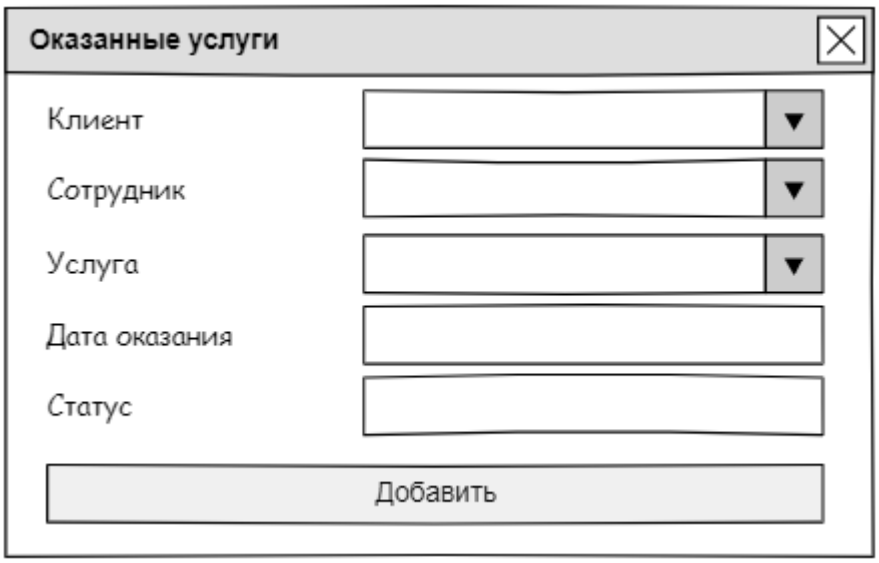

Рисунок 2.17. Макет окна создания/редактирования оказанных услуг

На рисунке 2.18 представлен макет окна «Поставщики», которое является основным рабочим экраном для администратора и менеджера по закупкам. Его цель — предоставить пользователю общий обзор всех поставщиков, с которыми работает ООО «Руссоль-центр», а также инструменты для их управления.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Поставщики» и кнопками управления (свернуть, развернуть, закрыть). В верхней части окна расположен блок поиска: поле ввода с меткой Поиск: позволяет быстро фильтровать список поставщиков по любому из столбцов (например, по наименованию компании или контактному лицу).

Основная часть: занимает большую часть окна и представляет собой таблицу со следующими столбцами:

- ID — уникальный номер записи о поставщике.
- Компания — наименование компании-поставщика.
- Контактное лицо — ФИО ответственного лица в компании.
- Номер телефона — контактный телефон поставщика.
- Электронная почта — адрес электронной почты для связи.
- Адрес — юридический или фактический адрес компании.

Ячейки таблицы заполняются данными из базы данных. Пользователь может просматривать данные, сортировать строки по любому столбцу (щелчком по заголовку) и выбирать строку для дальнейших действий.

Панель управления: В нижней части окна расположена панель с четырьмя основными кнопками:

- Добавить — открывает окно для создания новой записи о поставщике.
- Изменить — открывает окно редактирования выбранной записи.
- Удалить — удаляет выбранную запись (после подтверждения).
- Обновить — обновляет данные в таблице, отображая последние изменения в базе данных.

Этот макет обеспечивает удобное и эффективное управление справочником поставщиков, позволяя пользователю быстро находить нужные записи, просматривать их детали и выполнять основные операции (добавление, изменение, удаление) с помощью интуитивно понятных кнопок.

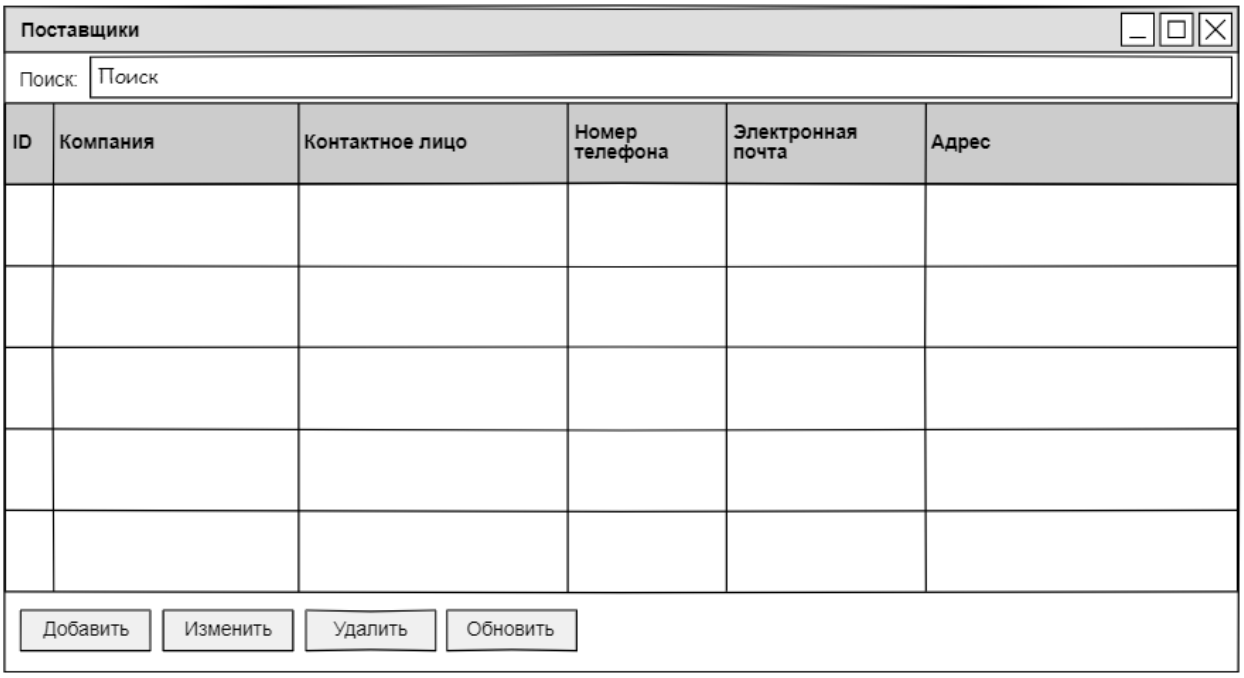

Рисунок 2.18. Макет окна поставщики

На рисунке 2.19 представлен макет окна, предназначенного для создания новой записи о поставщике или редактирования существующей в информационной системе ООО «Руссоль-центр». Это окно является

ключевым для администратора и менеджера по закупкам, позволяя им управлять справочником поставщиков.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Поставщики» и кнопкой закрытия. Внутри расположены пять полей ввода:

- Компания: Текстовое поле для ввода наименования компании-поставщика.
- Контактное лицо: Текстовое поле для ввода ФИО ответственного лица.
- Номер телефона: Текстовое поле для указания контактного телефона.
- Электронная почта: Текстовое поле для ввода адреса электронной почты.
- Адрес: Текстовое поле для указания юридического или фактического адреса компании.

Логика работы:

- При создании новой записи пользователь заполняет все поля и нажимает кнопку Добавить. После этого данные сохраняются в базе данных, и окно закрывается.
- При редактировании существующей записи окно открывается с уже подгруженными данными из базы данных. Пользователь может изменить любое поле (например, исправить номер телефона или адрес) и снова нажать добавить (в данном контексте это означает "Сохранить изменения"). Система обновляет запись в базе данных.
- Кнопка Добавить является универсальной: она используется как для создания новой записи, так и для сохранения изменений в существующей.

Этот макет обеспечивает простой и понятный интерфейс для управления справочником поставщиков, что является важным элементом для эффективного взаимодействия с внешними партнёрами и планирования закупок.

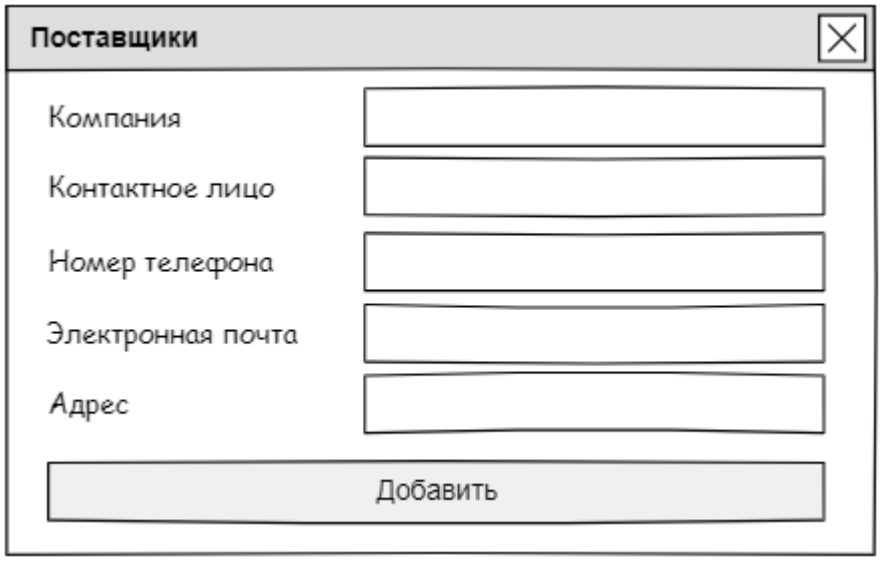

Рисунок 2.19. Макет окна создания/редактирования поставщика

На рисунке 2.20 представлен макет окна «Сотрудники», которое является основным рабочим экраном для администратора и генерального директора. Его цель — предоставить пользователю общий обзор всех сотрудников компании, а также инструменты для их управления.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Сотрудники» и кнопками управления (свернуть, развернуть, закрыть). В верхней части окна расположен блок поиска: поле ввода с меткой Поиск: позволяет быстро фильтровать список сотрудников по любому из столбцов (например, по ФИО или должности).

Основная часть: занимает большую часть окна и представляет собой таблицу со следующими столбцами:

- ID — уникальный номер записи о сотруднике.
- ФИО — фамилия, имя, отчество сотрудника.
- Должность — занимаемая сотрудником должность.
- Номер телефона — контактный телефон сотрудника.
- Электронная почта — адрес электронной почты для связи.
- Дата найма — дата, когда сотрудник был принят на работу.

Ячейки таблицы заполняются данными из базы данных. Пользователь может просматривать данные, сортировать строки по любому столбцу (щелчком по заголовку) и выбирать строку для дальнейших действий.

Панель управления: В нижней части окна расположена панель с четырьмя основными кнопками:

- Добавить — открывает окно для создания новой записи о сотруднике.
- Изменить — открывает окно редактирования выбранной записи.
- Удалить — удаляет выбранную запись (после подтверждения).
- Обновить — обновляет данные в таблице, отображая последние изменения в базе данных.

Этот макет обеспечивает удобное и эффективное управление справочником сотрудников, позволяя пользователю быстро находить нужные записи, просматривать их детали и выполнять основные операции (добавление, изменение, удаление) с помощью интуитивно понятных кнопок.

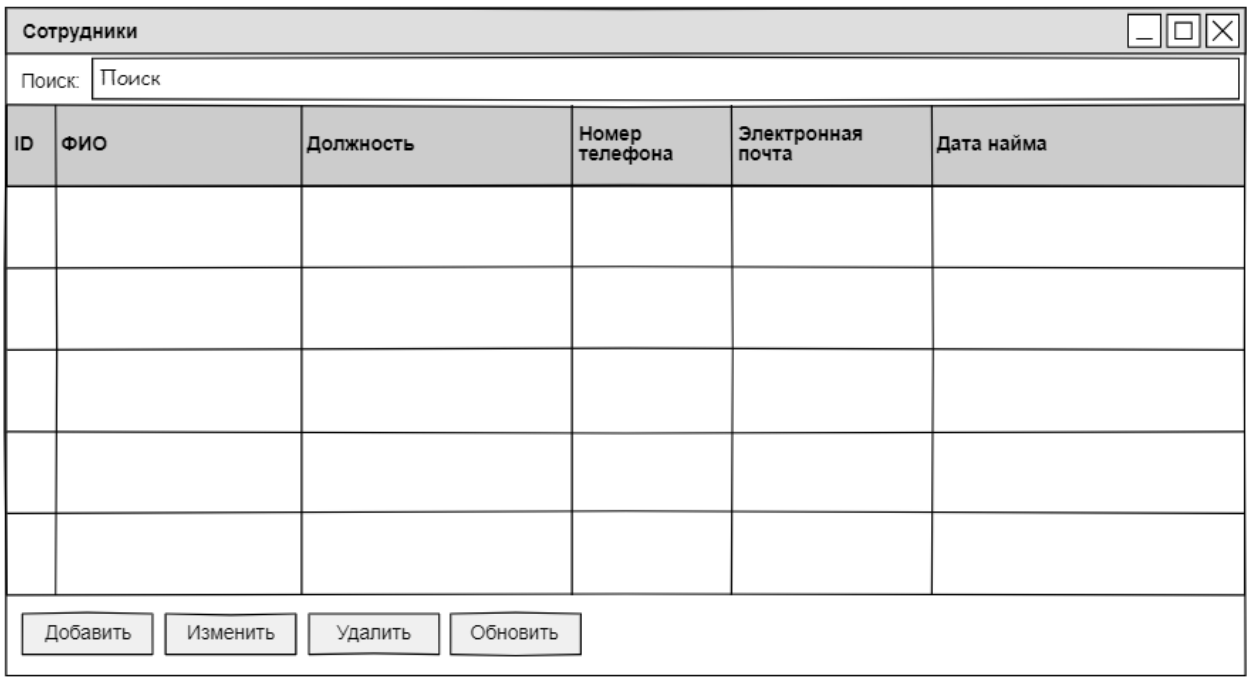

Рисунок 2.20. Макет окна сотрудники

На рисунке 2.21 представлен макет окна, предназначенного для создания новой записи о сотруднике или редактирования существующей в информационной системе ООО «Руссоль-центр». Это окно является ключевым для администратора и генерального директора, позволяя им управлять справочником персонала.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Сотрудник» и кнопкой закрытия. Внутри расположены пять полей ввода:

- ФИО: Текстовое поле для ввода фамилии, имени и отчества сотрудника.
- Должность: Текстовое поле для указания занимаемой должности.
- Номер телефона: Текстовое поле для ввода контактного номера телефона.
- Электронная почта: Текстовое поле для указания адреса электронной почты.
- Дата найма: Текстовое поле (или поле с календарём), в котором указывается дата, когда сотрудник был принят на работу.

Логика работы:

- При создании новой записи пользователь заполняет все поля и нажимает кнопку Добавить. После этого данные сохраняются в базе данных, и окно закрывается.
- При редактировании существующей записи окно открывается с уже подгруженными данными из базы данных. Пользователь может изменить любое поле (например, исправить номер телефона или обновить должность) и снова нажать добавить (в данном контексте это означает "Сохранить изменения"). Система обновляет запись в базе данных.
- Кнопка Добавить является универсальной: она используется как для создания новой записи, так и для сохранения изменений в существующей.

Этот макет обеспечивает простой и понятный интерфейс для управления данными о сотрудниках, что является важным элементом для эффективного управления персоналом компании.

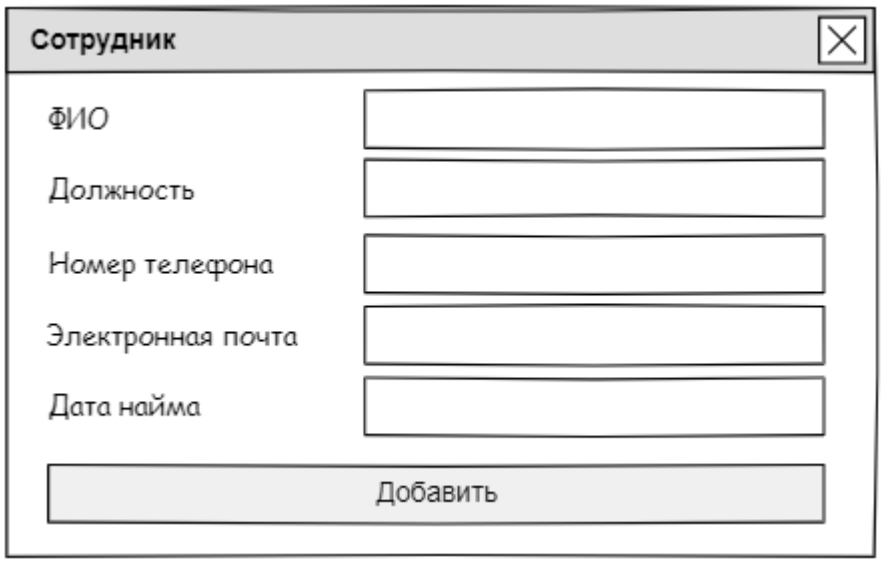

Рисунок 2.21. Макет окна создания/редактирования сотрудника

На рисунке 2.22 представлен макет окна «Авторизация», которое является административным интерфейсом для управления учётными записями пользователей системы. Это окно предназначено исключительно для администратора и позволяет ему просматривать, добавлять, изменять и удалять учётные записи, а также назначать им уровни доступа.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Авторизация» и кнопками управления (свернуть, развернуть, закрыть). В верхней части окна расположен блок поиска: поле ввода с меткой Поиск: позволяет быстро фильтровать список пользователей по любому из столбцов (например, по ФИО сотрудника или логину).

Основная часть: занимает большую часть окна и представляет собой таблицу со следующими столбцами:

- ID — уникальный номер записи об учётной записи.
- Сотрудник — ФИО сотрудника, которому привязана учётная запись. Ссылка на таблицу Сотрудники.
- Логин — имя пользователя, используемое для входа в систему.
- Пароль — пароль пользователя.
- Уровень доступа — роль пользователя в системе (Администратор, Менеджер по заказам, Менеджер по услугам), определяющая набор доступных функций.

Ячейки таблицы заполняются данными из базы данных. Пользователь может просматривать данные, сортировать строки по любому столбцу (щелчком по заголовку) и выбирать строку для дальнейших действий.

Панель управления: В нижней части окна расположена панель с четырьмя основными кнопками:

- Добавить — открывает окно для создания новой учётной записи (после выбора сотрудника и указания логина/пароля).
- Изменить — открывает окно редактирования выбранной учётной записи (изменение пароля или уровня доступа).
- Удалить — удаляет выбранную учётную запись (после подтверждения).
- Обновить — обновляет данные в таблице, отображая последние изменения в базе данных.

Этот макет обеспечивает удобное и эффективное управление правами доступа к системе, что является критически важным элементом для обеспечения безопасности и контроля над работой информационной системы ООО «Руссоль-центр».

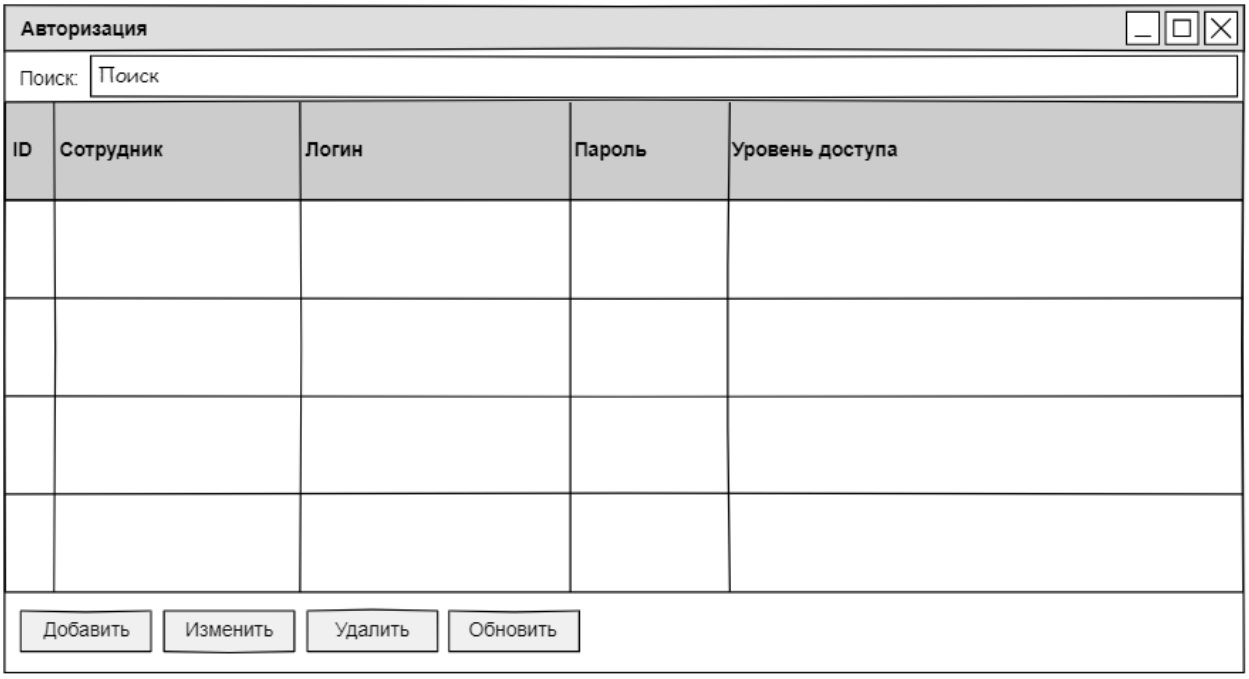

Рисунок 2.22. Макет окна авторизация пользователей

На рисунке 2.23 представлен макет окна, предназначенного для создания новой учётной записи пользователя или редактирования существующей в информационной системе ООО «Руссоль-центр». Это окно является ключевым для администратора и используется при назначении доступа новому сотруднику или изменении прав уже существующего.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Авторизация» и кнопкой закрытия. Внутри расположены четыре поля:

- Сотрудник: Поле с выпадающим списком (ComboBox), которое позволяет выбрать сотрудника из справочника. Это поле привязывает учётную запись к конкретному человеку.
- Логин: Текстовое поле для ввода уникального имени пользователя, которое будет использоваться для входа в систему.
- Пароль: Текстовое поле для указания пароля. Обычно символы пароля маскируются для безопасности.
- Уровень доступа: Поле с выпадающим списком, в котором выбирается роль пользователя (Администратор, Менеджер по заказам, Менеджер по услугам). Эта роль определяет набор функций, доступных пользователю в системе.

Логика работы:

- При создании новой учётной записи администратор выбирает сотрудника, задаёт логин и пароль, назначает уровень доступа и нажимает кнопку Добавить. После этого данные сохраняются в базе данных, и окно закрывается.
- При редактировании существующей записи окно открывается с уже подгруженными данными из базы данных. Администратор может изменить логин, пароль или уровень доступа и снова нажать добавить (в данном контексте это означает "Сохранить изменения"). Система обновляет запись в базе данных.
- Кнопка Добавить является универсальной: она используется как для создания новой записи, так и для сохранения изменений в существующей.

Этот макет обеспечивает простой и понятный интерфейс для управления правами доступа, что является критически важным элементом для обеспечения безопасности и контроля над работой информационной системы ООО «Руссоль-центр».

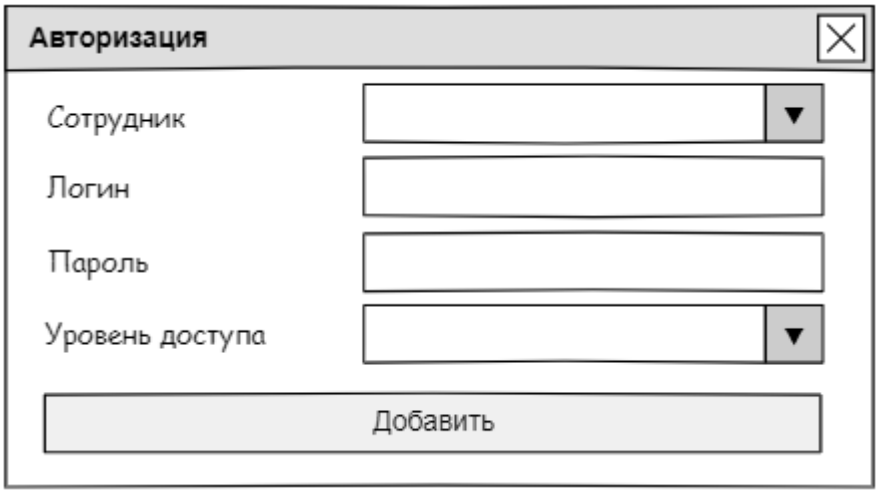

Рисунок 2.23. Макет окна создания/редактирования авторизация пользователя

На рисунке 2.24 представлен макет окна «Клиенты», которое является основным рабочим экраном для администратора, менеджера по заказам и генерального директора. Его цель — предоставить пользователю общий обзор всех клиентов компании, с которыми работает ООО «Руссоль-центр», а также инструменты для их управления.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Клиенты» и кнопками управления (свернуть, развернуть, закрыть). В верхней части окна расположен блок поиска: поле ввода с меткой Поиск: позволяет быстро фильтровать список клиентов по любому из столбцов (например, по наименованию организации или контактному лицу).

Основная часть: занимает большую часть окна и представляет собой таблицу со следующими столбцами:

- ID — уникальный номер записи о клиенте.
- Организация — наименование компании-клиента.
- Контактное лицо — ФИО ответственного лица в компании.
- Номер телефона — контактный телефон клиента.
- Электронная почта — адрес электронной почты для связи.

- Адрес — юридический или фактический адрес компании.

Ячейки таблицы заполняются данными из базы данных. Пользователь может просматривать данные, сортировать строки по любому столбцу (щелчком по заголовку) и выбирать строку для дальнейших действий.

Панель управления: В нижней части окна расположена панель с четырьмя основными кнопками:

- Добавить — открывает окно для создания новой записи о клиенте.
- Изменить — открывает окно редактирования выбранной записи.
- Удалить — удаляет выбранную запись (после подтверждения).
- Обновить — обновляет данные в таблице, отображая последние изменения в базе данных.

Этот макет обеспечивает удобное и эффективное управление справочником клиентов, позволяя пользователю быстро находить нужные записи, просматривать их детали и выполнять основные операции (добавление, изменение, удаление) с помощью интуитивно понятных кнопок.

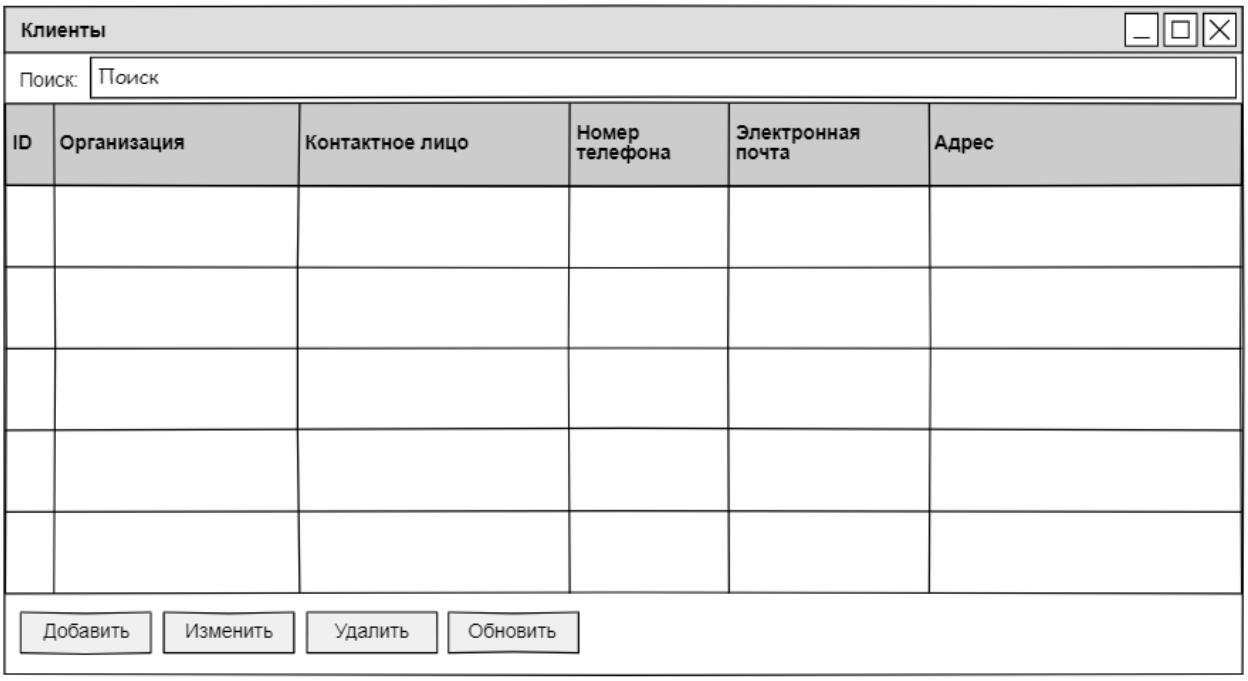

Рисунок 2.24. Макет окна клиенты

На рисунке 2.25 представлен макет окна, предназначенного для создания новой записи о клиенте или редактирования существующей в информационной системе ООО «Руссоль-центр». Это окно является

ключевым для администратора и менеджера по заказам, позволяя им управлять справочником клиентов.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Клиент» и кнопкой закрытия. Внутри расположены пять полей ввода:

- Организация: Текстовое поле для ввода наименования компании-клиента.
- Контактное лицо: Текстовое поле для ввода ФИО ответственного лица в компании.
- Номер телефона: Текстовое поле для указания контактного номера телефона.
- Электронная почта: Текстовое поле для ввода адреса электронной почты.
- Адрес: Текстовое поле для указания юридического или фактического адреса компании.

Логика работы:

- При создании новой записи пользователь заполняет все поля и нажимает кнопку Добавить. После этого данные сохраняются в базе данных, и окно закрывается.
- При редактировании существующей записи окно открывается с уже подгруженными данными из базы данных. Пользователь может изменить любое поле (например, исправить номер телефона или обновить адрес) и снова нажать добавить (в данном контексте это означает "Сохранить изменения"). Система обновляет запись в базе данных.
- Кнопка Добавить является универсальной: она используется как для создания новой записи, так и для сохранения изменений в существующей.

Этот макет обеспечивает простой и понятный интерфейс для управления данными о клиентах, что является важным элементом для эффективного взаимодействия с внешними партнёрами и оформления заказов.

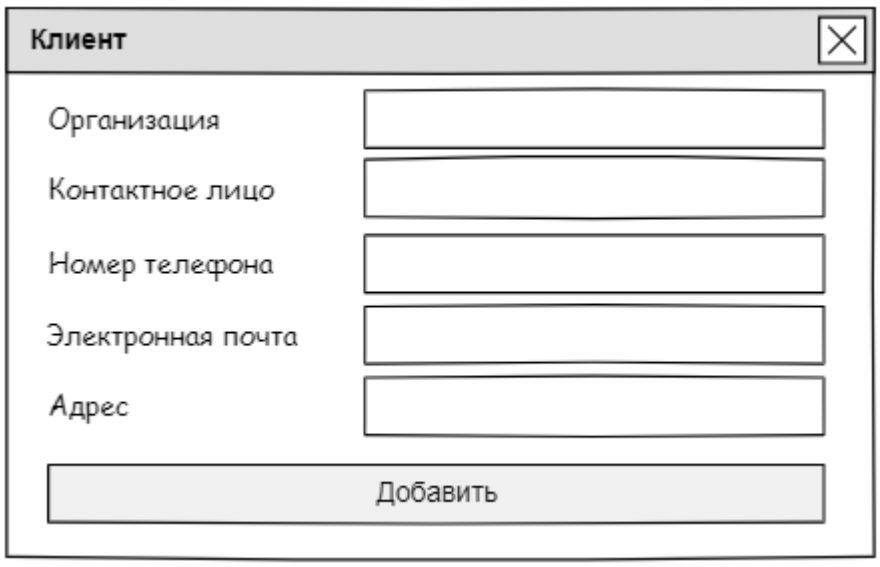

Рисунок 2.25. Макет окна создания/редактирования клиента

На рисунке 2.26 представлен макет окна «Товары», которое является основным рабочим экраном для администратора и менеджера по закупкам. Его цель — предоставить пользователю общий обзор всех товаров, которые предлагает ООО «Руссоль-центр», а также инструменты для их управления.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Товары» и кнопками управления (свернуть, развернуть, закрыть). В верхней части окна расположен блок поиска: поле ввода с меткой Поиск: позволяет быстро фильтровать список товаров по любому из столбцов (например, по наименованию или категории).

Основная часть: занимает большую часть окна и представляет собой таблицу со следующими столбцами:

- ID — уникальный номер записи о товаре.
- Поставщик — наименование компании-поставщика, от которой закупается данный товар.
- Наименование — название товара.
- Категория — принадлежность товара к определённой группе (например, "Соль пищевая", "Сода").
- Единица измерения — единица, в которой измеряется товар (кг, шт, л и т.д.).
- Цена за ед. товара — стоимость одной единицы товара.

- Описание — дополнительная информация о товаре.

Ячейки таблицы заполняются данными из базы данных. Пользователь может просматривать данные, сортировать строки по любому столбцу (щелчком по заголовку) и выбирать строку для дальнейших действий.

Панель управления: В нижней части окна расположена панель с четырьмя основными кнопками:

- Добавить — открывает окно для создания новой записи о товаре.
- Изменить — открывает окно редактирования выбранной записи.
- Удалить — удаляет выбранную запись (после подтверждения).
- Обновить — обновляет данные в таблице, отображая последние изменения в базе данных.

Этот макет обеспечивает удобное и эффективное управление справочником товаров, позволяя пользователю быстро находить нужные записи, просматривать их детали и выполнять основные операции (добавление, изменение, удаление) с помощью интуитивно понятных кнопок.

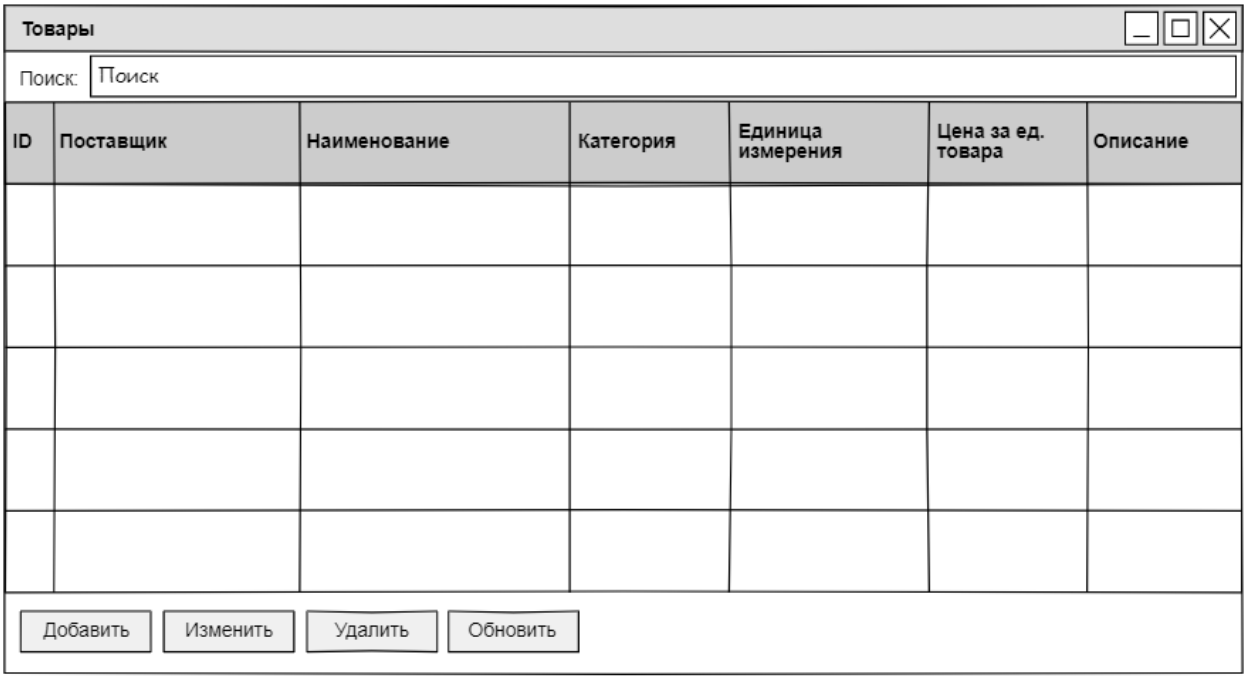

Рисунок 2.26. Макет окна товары

На рисунке 2.27 представлен макет окна, предназначенного для создания новой записи о товаре или редактирования существующей в информационной системе ООО «Руссоль-центр». Это окно является ключевым для

администратора и менеджера по закупкам, позволяя им управлять справочником номенклатуры.

Интерфейс: Окно имеет стандартную заголовочную панель с названием «Товар» и кнопкой закрытия. Внутри расположены шесть полей:

- Поставщик: Поле с выпадающим списком (ComboBox), которое позволяет выбрать компанию-поставщика из справочника.
- Наименование: Текстовое поле для ввода названия товара.
- Категория: Текстовое поле для указания категории товара (например, "Соль пищевая", "Сода").
- Единица измерения: Текстовое поле для ввода единицы, в которой измеряется товар (кг, шт, л и т.д.).
- Цена за ед. товара: Текстовое поле для указания стоимости одной единицы товара.
- Описание: Текстовое поле для ввода дополнительной информации о товаре.

Логика работы:

- При создании новой записи пользователь заполняет все поля и нажимает кнопку Добавить. После этого данные сохраняются в базе данных, и окно закрывается.
- При редактировании существующей записи окно открывается с уже подгруженными данными из базы данных. Пользователь может изменить любое поле (например, исправить цену или обновить описание) и снова нажать добавить (в данном контексте это означает "Сохранить изменения"). Система обновляет запись в базе данных.
- Кнопка Добавить является универсальной: она используется как для создания новой записи, так и для сохранения изменений в существующей.

Этот макет обеспечивает простой и понятный интерфейс для управления данными о товарах, что является важным элементом для эффективного планирования закупок и оформления заказов.

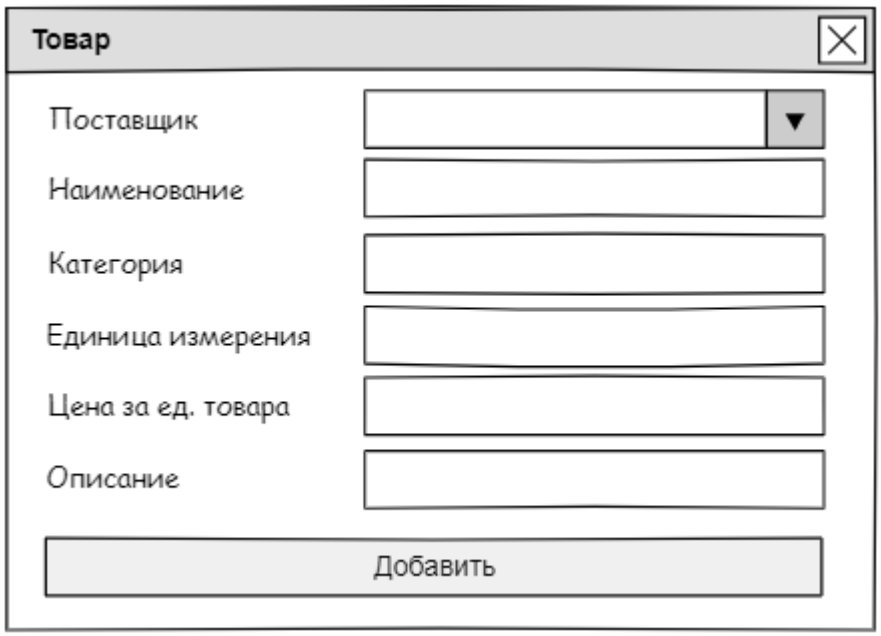

Рисунок 2.27. Макет окна создания/редактирования товара

2.5. Описание логики функционирования ПО

Разработанное программное обеспечение информационной системы для ООО «Руссоль-центр» представляет собой десктопное приложение, реализующее комплексную автоматизацию ключевых бизнес-процессов предприятия оптовой торговли. Логика функционирования ПО направлена на обеспечение эффективного взаимодействия между пользователями системы (Администратор, Менеджер по заказам, Менеджер по услугам) и хранимыми в базе данных сведениями о клиентах, поставщиках, товарах, заказах, оказанных услугах и сотрудниках. Система построена по клиент-серверной архитектуре, где клиентская часть (представленная десктопным приложением) отвечает за предоставление пользовательского интерфейса и взаимодействие с пользователем, а серверная часть — за хранение и обработку данных (в данном случае, СУБД Microsoft SQL Server, расположенная на локальном сервере/рабочей станции). Взаимодействие между интерфейсом и базой данных осуществляется через специализированные модули приложения, обеспечивающие выполнение CRUD-операций (Create, Read, Update, Delete) и обработку бизнес-логики.

2.5.1. Общая архитектура и компоненты ПО

Программное обеспечение состоит из следующих основных компонентов:

1) Пользовательский интерфейс (UI): Реализован с использованием Windows Forms в среде .NET Framework/C#. Включает в себя окна авторизации, главное меню, окна для работы с заказами, услугами, справочниками (товары, клиенты, сотрудники, поставщики) и отчётами. Интерфейс разработан с учётом требований к удобству использования (Usability), обеспечивая интуитивно понятную навигацию и логичную структуру элементов управления. Макеты интерфейса, описанные в разделе 2.4, определяют внешний вид и логику взаимодействия с каждым окном.

2) Слой приложения (Application Layer): включает в себя логику, управляющую работой системы. Он отвечает за обработку команд пользователя, вызов соответствующих методов для работы с данными, выполнение бизнес-правил (например, проверка наличия товара на складе при создании заказа, расчёт суммы заказа) и формирование ответа для отображения в UI. Этот слой может включать классы, отвечающие за управление сессией пользователя, валидацию вводимых данных и т.д.

3) Слой доступа к данным (Data Access Layer): Этот компонент реализует взаимодействие с базой данных SQL Server. Он содержит методы для выполнения SQL-запросов (SELECT, INSERT, UPDATE, DELETE), получения и передачи данных между базой данных и слоем приложения. Используются технологии ADO.NET или ORM (например, Entity Framework), обеспечивающие эффективное и безопасное обращение к данным.

4) База данных: Физическая реализация модели данных. Содержит таблицы, связи, индексы и ограничения, обеспечивающие целостность и производительность хранения информации.

2.5.2. Вход в систему и разграничение прав

Логика функционирования системы начинается с процесса аутентификации пользователя.

1) При запуске приложения открывается окно авторизации (см. рис. 2.6). Пользователь вводит свои Логин и Пароль.

2) Приложение обращается к базе данных, проверяя введённые учётные данные в таблице, связанной с пользователями (например, Employees и Auth). Пароли хранятся в зашифрованном виде.

3) Если данные верны, система извлекает информацию о роли пользователя (например, Администратор, Менеджер по заказам, Менеджер по услугам).

4) После успешной аутентификации открывается главное окно (см. рис. 2.7). Содержимое главного окна (набор видимых и доступных кнопок) динамически изменяется в зависимости от роли вошедшего пользователя. Это обеспечивает разграничение доступа на уровне интерфейса. Например, менеджер по заказам увидит кнопки Заказы, Клиенты, Склад, Товары, но не увидит Сотрудники или Услуги (если они не входят в его функционал), а администратор — все кнопки.

2.5.3. Работа с заказами

Модуль управления заказами является центральным для менеджера по заказам.

1) При нажатии кнопки Заказы в главном меню открывается окно списка заказов (см. рис. 2.10). Система выполняет SQL-запрос для извлечения данных из таблицы Orders и отображает их в виде таблицы.

2) Пользователь может использовать фильтр поиска для нахождения конкретного заказа.

3) Для создания нового заказа пользователь нажимает кнопку Добавить. Открывается окно редактирования заказа (см. рис. 2.11) с пустыми полями (или с предустановленной датой).

4) Пользователь выбирает Клиента и Сотрудника из выпадающих списков (данные подгружаются из соответствующих таблиц Clients и Employees). Указывает Дата заказа, Сумма рассчитывается автоматически на

основе выбранных товаров, Статус устанавливается по умолчанию (например, "Новый").

5) В нижней таблице Товары пользователь добавляет позиции: выбирает товар из списка (данные из Products) и указывает Количество. При выборе товара система может проверять его наличие на складе (Stock).

6) После заполнения всех данных пользователь нажимает кнопку Добавить. Приложение формирует SQL-запрос INSERT для добавления новой записи в таблицу Orders, а затем INSERT для каждой позиции в таблицу OrderItems, а также, при необходимости, обновляет остатки в таблице Stock.

7) При редактировании существующего заказа (выбор строки в списке и нажатие изменить) окно открывается с уже подгруженными данными (SELECT запросы). Пользователь вносит изменения и снова нажимает. Добавить, что вызывает выполнение SQL-запроса UPDATE для соответствующих записей в Orders и OrderItems.

8) Операции. Удалить и обновить в окне списка также инициируют выполнение соответствующих SQL-запросов DELETE и SELECT для обновления отображаемых данных.

2.5.4. Работа с оказанными услугами

Модуль управления услугами функционирует аналогично модулю заказов, но ориентирован на менеджера по услугам.

1) Окно Оказанные услуги отображает список услуг из таблицы ProvidedServices.

2) При добавлении/редактировании (см. рис. 2.17) пользователь выбирает Клиента, Сотрудника, Услугу (из справочника Services), указывает Дата оказания и Статус.

3) При сохранении (Добавить) данные записываются в таблицу ProvidedServices (INSERT или UPDATE).

2.5.5. Работа со справочниками и складом

Для администратора и других авторизованных пользователей доступны окна для управления справочниками (Клиенты, Поставщики, Товары,

Сотрудники). Логика работы с ними схожа: отображение списка, возможность поиска, добавления (INSERT), изменения (UPDATE) и удаления (DELETE) записей.

Модуль Склад (см. рис. 2.14) отображает остатки товаров (данные из Stock и Products). Пользователь может добавить новую запись о товаре на складе или изменить его количество, что приводит к обновлению соответствующей записи в таблице Stock.

Логика функционирования разработанного ПО обеспечивает полный цикл управления бизнес-процессами, от приёма заказа клиента до формирования отчётов. Она чётко структурирована, разделяет уровни доступа и реализует все основные функции. Архитектура приложения позволяет эффективно управлять данными, минимизировать ручной труд и повысить общую эффективность работы ООО «Руссоль-центр».

3. СПЕЦИАЛЬНАЯ ЧАСТЬ

3.1. Формирование требований к ПО

Формирование требований к программному обеспечению (ПО) является критически важным этапом в процессе разработки информационной системы для ООО «Руссоль-центр». Он представляет собой систематический подход к определению и документированию того, что система должна делать и как она должна это делать, чтобы соответствовать целям автоматизации, удовлетворять потребности пользователей и бизнес-процессы предприятия. От качества и полноты сформулированных требований зависит успешность последующих этапов: проектирования, разработки, тестирования и внедрения. Недостаточно точные или неполные требования могут привести к созданию системы, не отвечающей ожиданиям, увеличению сроков и стоимости проекта, а также к сложностям при эксплуатации.

Целью формирования требований в рамках дипломного проекта является создание чёткого и однозначного технического задания, который служит основой для всех участников процесса разработки. Для разработчиков он определяет, какие функции необходимо реализовать и какими характеристиками должна обладать система. Для тестировщиков — критерии оценки соответствия готового продукта спецификации. Для заказчика — документ, подтверждающий, что создаваемое ПО решает поставленные задачи. Этот этап помогает избежать недопонимания между командой разработчиков и бизнесом, гарантируя, что финальный продукт действительно будет полезным инструментом для оптимизации деятельности ООО «Руссоль-центр».

Процесс формирования требований опирается на результаты предыдущих этапов проекта. В первую очередь, это анализ предметной, где была рассмотрена структура предприятия, его ключевые бизнес-процессы (управление заказами, складом, взаимодействие с поставщиками и клиентами), а также выявлены проблемы, требующие автоматизации. Также важную роль сыграли результаты анализа систем-аналогов, которые

позволили понять, какие функции уже реализованы в существующих решениях, и почему для специфической деятельности ООО «Руссоль-центр» может потребоваться индивидуальная разработка. Наконец, результаты проектирования — такие как ER-диаграмма, Use Case диаграммы и макеты интерфейса — также повлияли на уточнение и структурирование требований.

Требования к ПО условно делятся на функциональные и нефункциональные. Функциональные требования описывают что система должна делать — какие действия она выполняет, какие сервисы предоставляет. Они связаны с конкретными операциями, которые могут инициировать пользователи или внешние системы. Для информационной системы ООО «Руссоль-центр» к функциональным требованиям относятся:

- Возможность управления заказами клиентов: создание, редактирование, просмотр статуса заказа, формирование печатных форм (счетов, накладных). Это касается как менеджеров по заказам, так и администраторов.
- Возможность управления оказанными услугами: фиксация факта оказания услуги, привязка к клиенту и сотруднику, изменение статуса, формирование актов. Это функционал для менеджеров по услугам и администраторов.
- Управление справочниками: работа с данными о клиентах, поставщиках, товарах, сотрудниках. Возможность просмотра, поиска, фильтрации, а для администратора — добавления, редактирования и удаления записей.
- Управление складом: отображение актуальных остатков, формирование заданий на сборку (для администратора), учёт операций (для администратора).
- Финансовый учёт и отчёты: фиксация оплат, формирование отчётов по продажам, услугам, клиентам, остаткам, возможный экспорт отчётов. Это касается как менеджеров, так и бухгалтеров и руководства.

- Управление пользователями и безопасностью: регистрация, аутентификация, разграничение прав доступа по ролям (Администратор, Менеджер по заказам, Менеджер по услугам), ведение журнала аудита (для администратора).

Нефункциональные требования, в свою очередь, описывают *как* система должна работать, определяя её характеристики и ограничения, не связанные напрямую с конкретными функциями, но критически важные для её успешного функционирования и принятия пользователями. К ним относятся:

- Производительность: система должна отвечать на запросы пользователя в разумные сроки (например, до 2 секунд) при работе нескольких пользователей одновременно.
- Надёжность: система должна быть устойчивой к сбоям и обеспечивать целостность данных при корректной эксплуатации. Обязательна реализация резервного копирования данных.
- Безопасность: реализация аутентификации и авторизации пользователей, шифрование паролей, разграничение доступа к данным и функциям в зависимости от роли пользователя.
- Удобство использования (Usability): интерфейс должен быть интуитивно понятным, соответствовать общепринятым стандартам, быть локализован на русский язык.
- Совместимость: система должна корректно работать в предполагаемой операционной системе (например, Windows).
- Масштабируемость: архитектура системы должна предусматривать возможность добавления новых функций и увеличения числа пользователей в будущем.

Таким образом, процесс формирования требований к ПО для ООО «Руссоль-центр» включает в себя комплексный анализ всех аспектов взаимодействия пользователей с системой и её работы в целом. Он обеспечивает чёткое понимание того, какой функционал должна реализовывать система, какими характеристиками она должна обладать и

каким образом она будет соответствовать целям автоматизации бизнес-процессов предприятия. Эти требования становятся основой для всех последующих этапов разработки, обеспечивая направление и критерии оценки качества создаваемого программного продукта.

3.2. Разработка интерфейса ПО

Разработка пользовательского интерфейса (UI) является одной из ключевых задач в процессе создания информационной системы для ООО «Руссоль-центр». На этом этапе воплощаются в жизнь требования к функциональности и удобству использования, сформулированные на предыдущих этапах проектирования (см. разделы 2.1, 2.4). Целью разработки интерфейса является создание интуитивно понятного, эффективного и удобного средства взаимодействия между пользователем и логикой приложения, что напрямую влияет на продуктивность работы сотрудников и общую удовлетворённость от использования системы.

В рамках главы 3.2 описывается процесс создания визуальных элементов и логики взаимодействия для основных функций системы, таких как управление заказами, оказанными услугами, справочниками (товары, клиенты, сотрудники, поставщики) и отчётами. Работа основывается на ранее созданных макетах (см. раздел 2.4), которые послужили основой для программной реализации. Здесь подробно рассматриваются выбранные элементы управления, структура окон, реализация навигации и способы ввода/вывода данных, обеспечивая преемственность между проектным дизайном и финальным программным продуктом.

Ниже представлены окна программы, созданные по макетам (см. раздел 2.4).

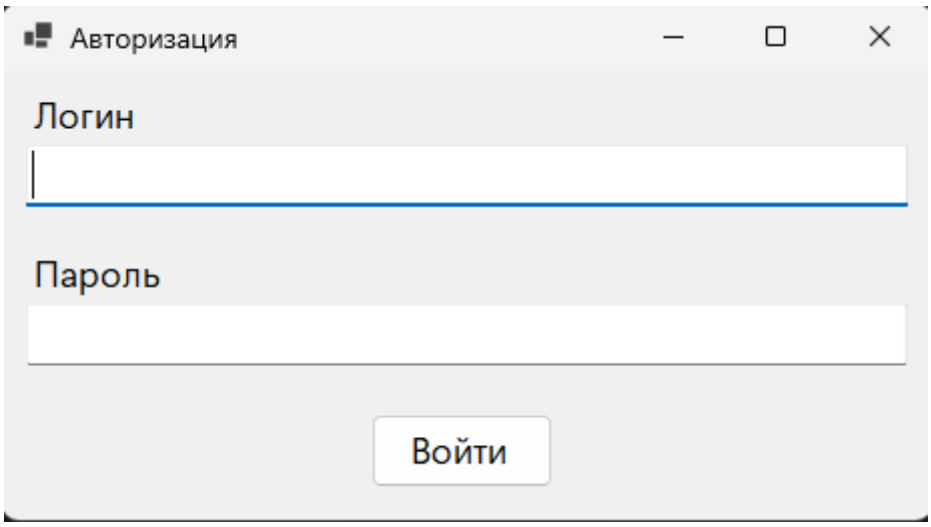

Рисунок 3.1. Окно авторизации

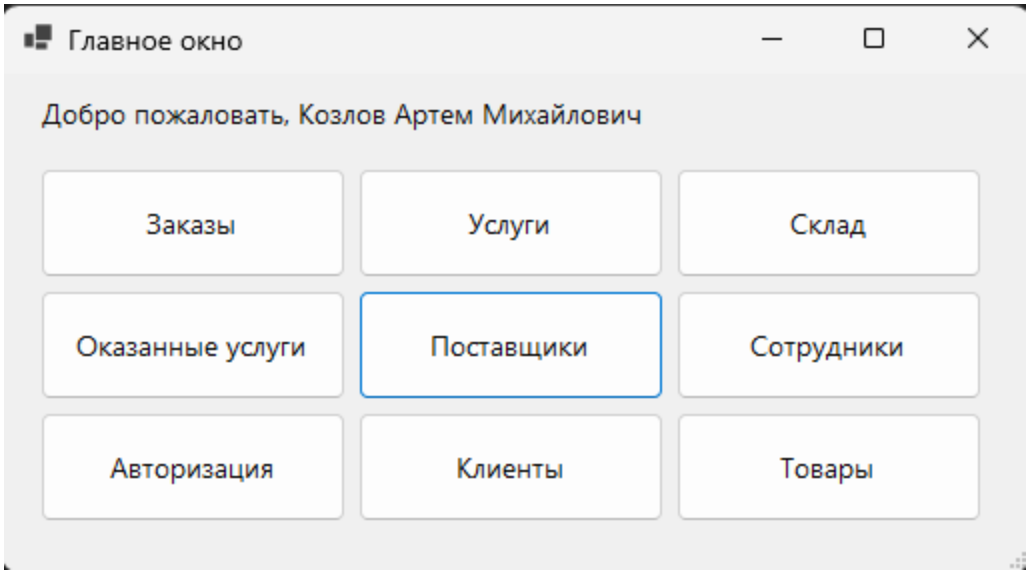

Рисунок 3.2. Главное окно администратора

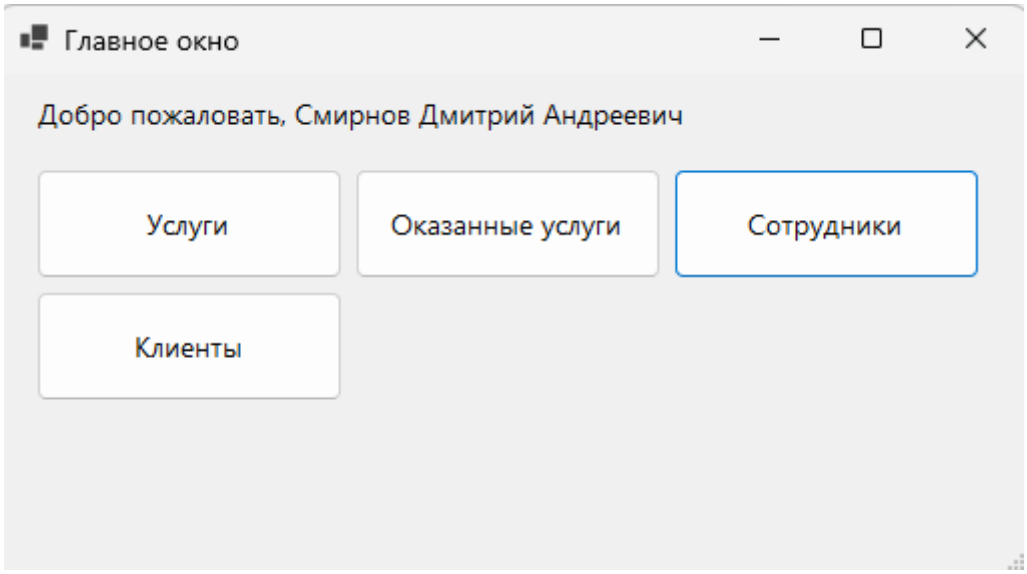

Рисунок 3.3. Главное окно менеджера по услугам

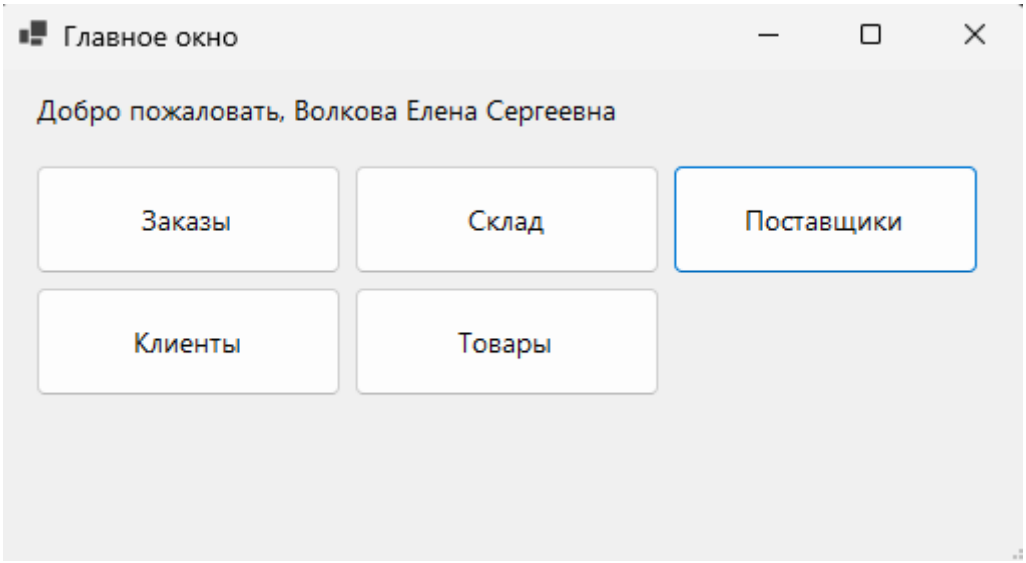

Рисунок 3.4. Главное окно менеджера по заказам

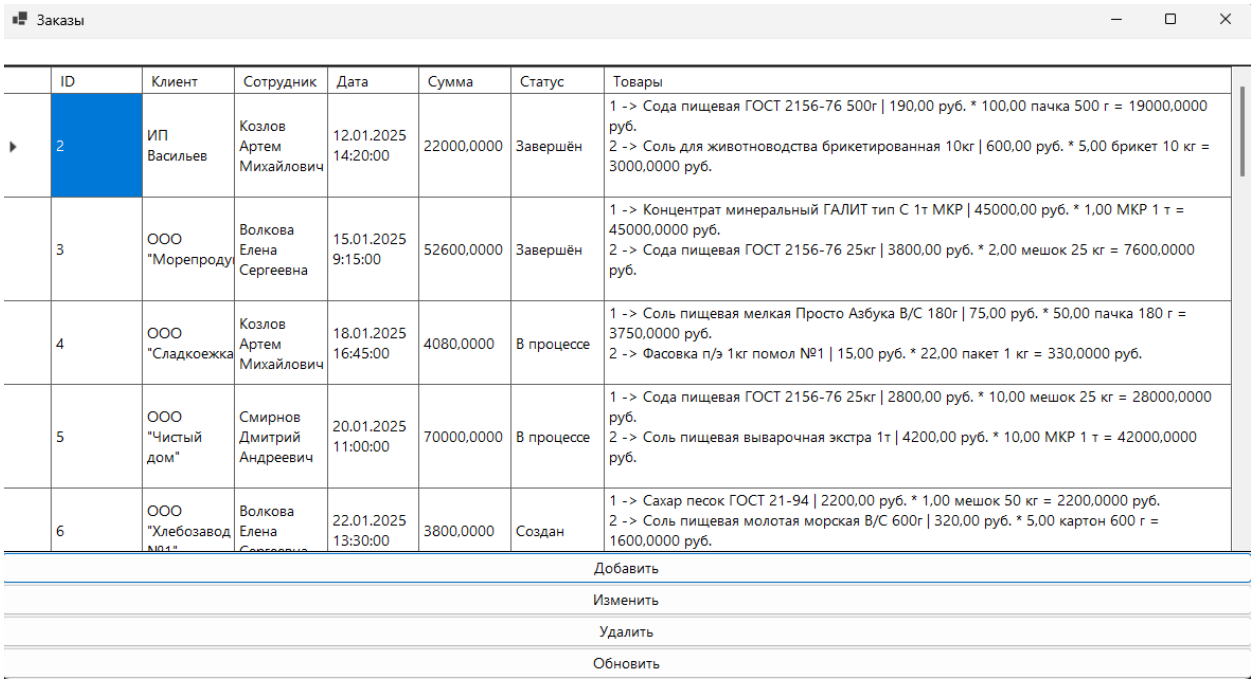

Рисунок 3.5. Окно просмотра заказов

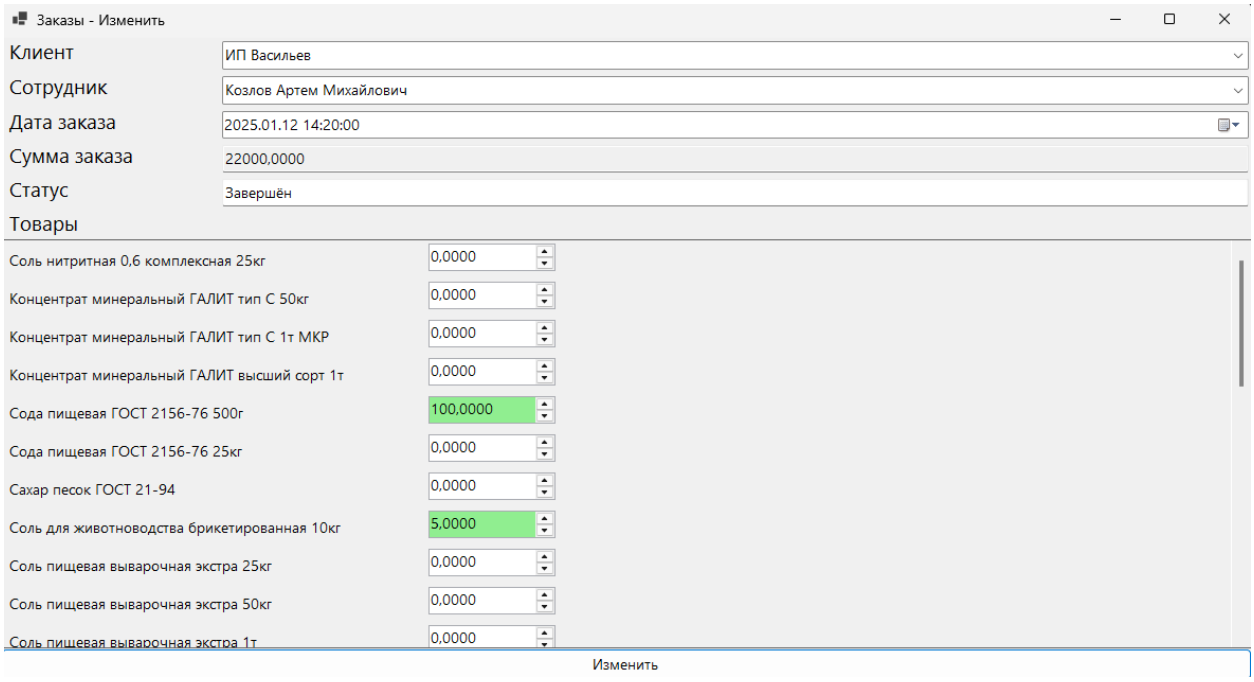

Рисунок 3.6. Окно изменения заказа

Изменение товаров в заказе происходит с помощью самописного элемента (UserControl). Это список (ListBox) с элементами (LabeledNumericItem) состоящего из текста (Label) и цифрового блока (NumericUpDown). При генерации элементов в список им присваиваются:

ID – идентификатор товара

- Key – наименование товара (то что выводится в Label)
- Value – значение, сколько в заказе (по умолчанию 0)
- Maximum – значение ограничивающее NumericUpDown по количества товара на складе
- Action – ивент вызываемый при взаимодействии с NumericUpDown (нужен для изменения цветового индикатора)

При открытии окна изменения содержимого заказа в элементы подгружаются данные из БД, при этом для наглядности что входит в заказ элементы заказа подсвечены зелёным. А при отсутствии товара на складе и, следовательно, невозможностью его добавить в заказ NumericUpDown отключается и становиться красного цвета.

Услуги

ID	Наименование	Описание	Цена
1	Доставка до склада клиента	Транспортировка заказа до указанного адреса	5000,00
2	Консультация по подбору соли	Профессиональная консультация технолога	2000,00
3	Подготовка сертификатов качества	Оформление документов на продукцию	1500,00
4	Гибкая упаковка под заказ	Фасовка продукции по индивидуальным мешкам	3000,00
5	Аналитика состава	Лабораторный анализ образца соли	4000,00
6	Обучение персонала	Инструктаж по работе с солью и добавками	2500,00
7	Разработка рецептуры	Создание формулы с использованием соли	3500,00
8	Контроль качества при получении	Проверка партии на соответствие ГОСТ	1800,00
9	Упаковка в брикеты	Прессование соли в брикеты 10 кг	1200,00
10	Фасовка в п/э пакеты	Разделение на упаковки по 1 кг	800,00
11	Маркировка продукции	Нанесение этикеток согласно ТР ТС	600,00
12	Создание онлайн-каталога	Размещение товаров в цифровом виде	7000,00
13	Поддержка API интеграции	Настройка обмена данными с ERP системами	10000,00
14	Резервирование партии	Бронирование товара на складе	500,00

Добавить

Изменить

Удалить

Обновить

Рисунок 3.7. Окно просмотра услуг

Услуги - Изменить

Наименование: Доставка до склада клиента

Описание: Транспортировка заказа до указанного адреса

Цена: 5000,00

Изменить

Рисунок 3.8. Окно изменения данных об услуге

Склад

ID	Товар	Количество на складе	Дата изменения
1	Соль нитритная 0,6 комплексная 25кг	170,01	26.10.2025 21:48:35
2	Концентрат минеральный ГАЛИТ тип С 50кг	80,00	26.10.2025 21:48:35
3	Концентрат минеральный ГАЛИТ тип С 1т МКР	7,00	26.10.2025 21:48:35
4	Концентрат минеральный ГАЛИТ высший сорт 1т	53,00	26.10.2025 21:48:35
5	Сода пищевая ГОСТ 2156-76 500г	300,00	26.10.2025 21:48:35
6	Сода пищевая ГОСТ 2156-76 25кг	40,00	26.10.2025 21:48:35
7	Сахар песок ГОСТ 21-94	60,00	26.10.2025 21:48:35
8	Соль для животноводства брикетированная 10кг	75,00	26.10.2025 21:48:35
9	Соль пищевая выварочная экстра 25кг	150,00	26.10.2025 21:48:35
10	Соль пищевая выварочная экстра 50кг	80,00	26.10.2025 21:48:35
11	Соль пищевая выварочная экстра 1т	5,00	26.10.2025 21:48:35
12	Соль пищевая выварочная экстра 1кг бумажный пакет	100,00	26.10.2025 21:30:20
13	Соль пищевая выварочная экстра 0,25кг п/п туба	200,00	26.10.2025 21:30:20
14	Соль пищевая выварочная экстра 0,5кг п/п туба	180,00	26.10.2025 21:30:20

Добавить

Изменить

Удалить

Обновить

Рисунок 3.9. Окно просмотра содержимого склада

Склад - Изменить

Товар: Соль нитритная 0,6 комплексная 25кг

Количество на: 170,01

Изменить

Рисунок 3.10. Окно изменения записей о содержимом склада

Оказанные услуги

ID	Клиент	Сотрудник	Услуга	Дата оказания	Статус
1	ООО "Солёный хлеб"	Козлов Артем Михайлович	Доставка до склада клиента	11.01.2025 11:00:00	Завершён
2	ИП Васильев	Волкова Елена Сергеевна	Консультация по подбору соли	13.01.2025 15:30:00	Завершён
3	ООО "Морепродукт"	Волкова Елена Сергеевна	Консультация по логистике	16.01.2025 10:00:00	Завершён
4	ООО "Сладкоежка"	Козлов Артем Михайлович	Гибкая упаковка под заказ	19.01.2025 14:00:00	В процессе
5	ООО "Чистый дом"	Петрова Ольга Николаевна	Аналитика состава	21.01.2025 12:00:00	Создан
6	ООО "Хлебозавод №1"	Волкова Елена Сергеевна	Доставка до склада клиента	23.01.2025 9:30:00	Завершён
7	ООО "Мясокомбинат Восток"	Козлов Артем Михайлович	Консультация по подбору соли	26.01.2025 11:15:00	Завершён
8	ООО "Супермаркет Юг"	Иванов Максим Львович	Подготовка сертификатов качества	28.01.2025 16:00:00	Создан
9	ООО "Фудсервис"	Смирнов Дмитрий Андреевич	Гибкая упаковка под заказ	29.01.2025 10:00:00	Создан
10	ООО "Кейтеринг Плюс"	Козлов Артем Михайлович	Аналитика состава	30.01.2025 13:00:00	Новый
11	ООО "Сеть Кафе"	Волкова Елена Сергеевна	Обучение персонала	01.02.2025 10:00:00	Новый
12	ООО "Пицца Мания"	Смирнов Дмитрий Андреевич	Разработка рецептуры	02.02.2025 11:00:00	В процессе
13	ООО "Кондитерская №1"	Козлов Артем Михайлович	Контроль качества при получении	03.02.2025 12:00:00	Создан
14	ООО "Сырный рай"	Петрова Ольга Николаевна	Упаковка в брикеты	04.02.2025 13:00:00	Завершён

Добавить

Изменить

Удалить

Обновить

Рисунок 3.11. Окно просмотра оказанных услуг

Оказанные услуги - Изменить

Клиент: ООО "Солёный хлеб"

Сотрудник: Козлов Артем Михайлович

Услуга: Доставка до склада клиента

Дата оказания: 2025.01.11 11:00:00

Статус: Завершён

Изменить

Рисунок 3.12. Окно изменения данных об оказанных услугах

Поставщики

ID	Компания	Контактное лицо	Номер телефона	Электронная почта	Адрес
1	ООО РУССОЛЬ	Иван Петров	+7 (495) 123-45-67	info@russol.ru	г. Новомосковск, ул. Соляная, 1
2	ЗАО Илецкий карьер	Марина Сидорова	+7 (345) 234-56-78	zakaz@ileck.ru	г. Илецк, шахта №3
3	ООО Мясной мир	Алексей Кузнецов	+7 (499) 345-67-89	supply@myasnoy.ru	Москва, пр-д Колбасный, 5
4	ООО Сладкая жизнь	Елена Федорова	+7 (812) 456-78-90	sugar@sladko.ru	Санкт-Петербург, ул. Конфетная, 10
5	ООО ХимПродукт	Дмитрий Морозов	+7 (411) 567-89-01	chem@soda.ru	Новосибирск, химзавод, 2
6	ООО Солерудник	Ольга Волкова	+7 (495) 222-33-44	geolog@solerud.ru	Тульская область, п. Галитный
7	ООО Белый кристалл	Сергей Попов	+7 (845) 333-44-55	sales@whitecrystal.ru	Саратов, ул. Промышленная, 12
8	ООО ТоргСоль	Наталья Крылова	+7 (473) 444-55-66	trade@torgsol.ru	Воронеж, ул. Складская, 8
9	ООО Южсолепродукт	Андрей Михайлов	+7 (861) 555-66-77	south@saltprod.ru	Краснодар, шоссе Нефтяников, 15
10	ООО УралХимСоль	Евгений Орлов	+7 (343) 666-77-88	ural@khim-sol.ru	Екатеринбург, пр-кт Ленина, 40
11	ООО Северсоль	Людмила Нестерова	+7 (818) 777-88-99	north@seversol.ru	Архангельск, ул. Полярная, 5
12	ООО АгроСоль	Виктор Титов	+7 (846) 888-99-00	agro@agrosol.ru	Самара, с. Солёное
13	ООО КавказСоль	Руслан Магомедов	+7 (878) 999-00-11	kavkaz@kavsol.ru	Пятигорск, ул. Горная, 3

Добавить

Изменить

Удалить

Обновить

Рисунок 3.13. Окно просмотра данных о поставщиках

Поставщики - Изменить

Компания: ООО РУССОЛЬ

Контактное лицо: Иван Петров

Номер телефона: +7 (495) 123-45-67

Электронная почта: info@russol.ru

Адрес: г. Новомосковск, ул. Соляная, 1

Изменить

Рисунок 3.14. Окно изменения данных о поставщике

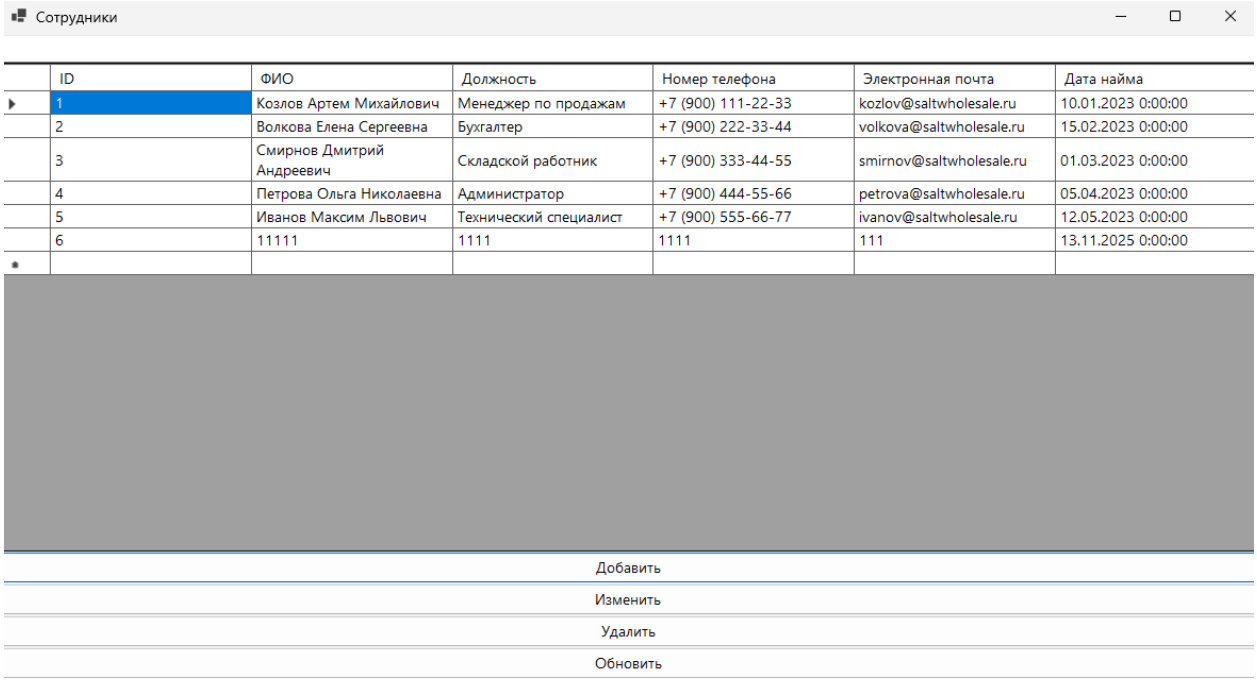

Рисунок 3.15. Окно просмотра данных сотрудников

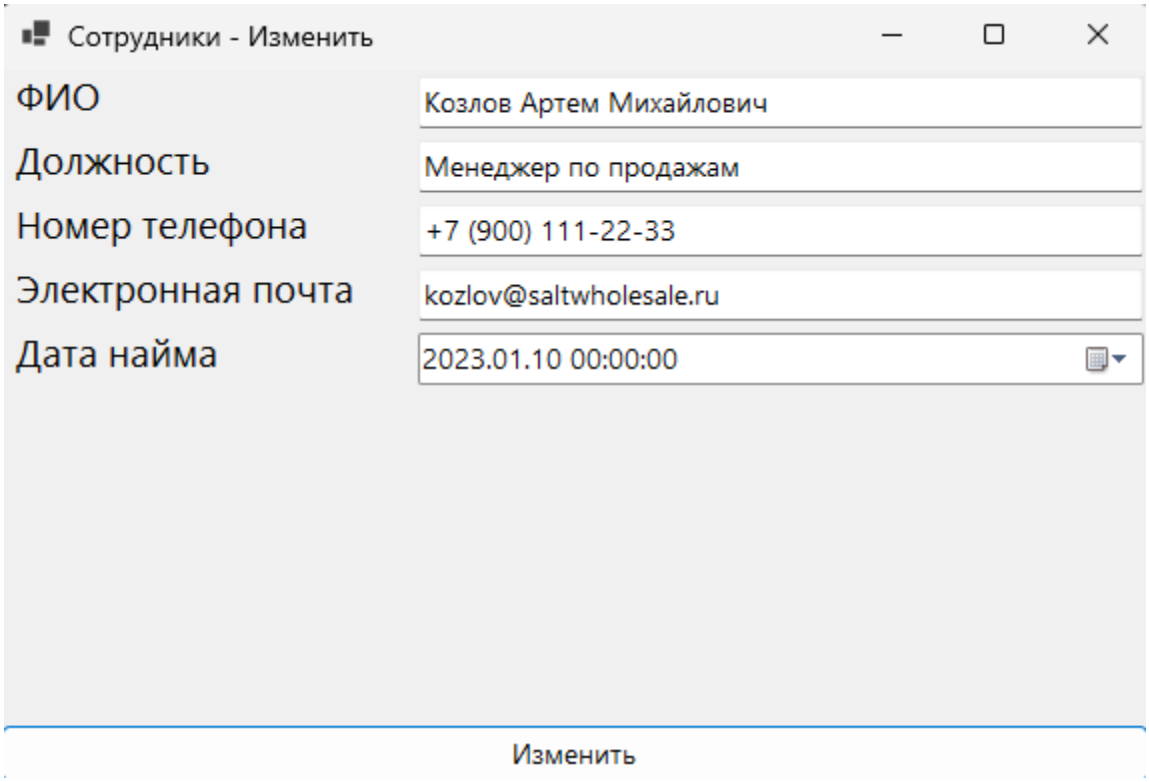

Рисунок 3.16. Окно изменения данных сотрудника

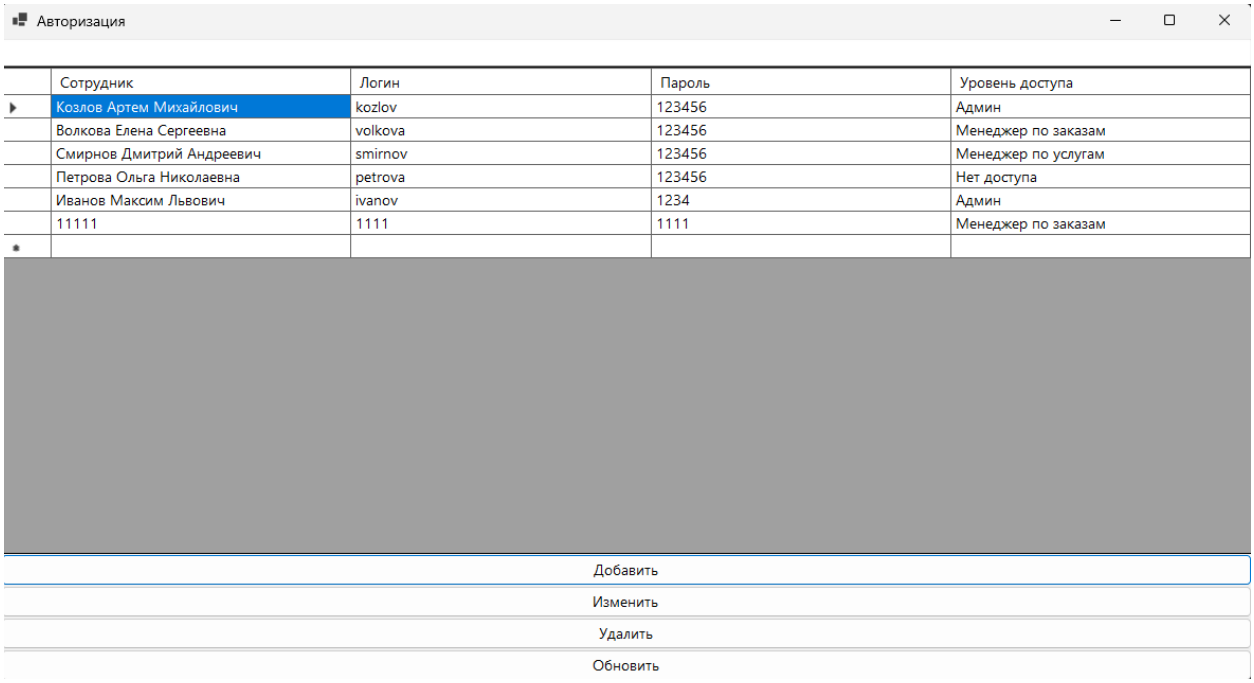

Рисунок 3.17. Окно просмотра данных авторизации сотрудников

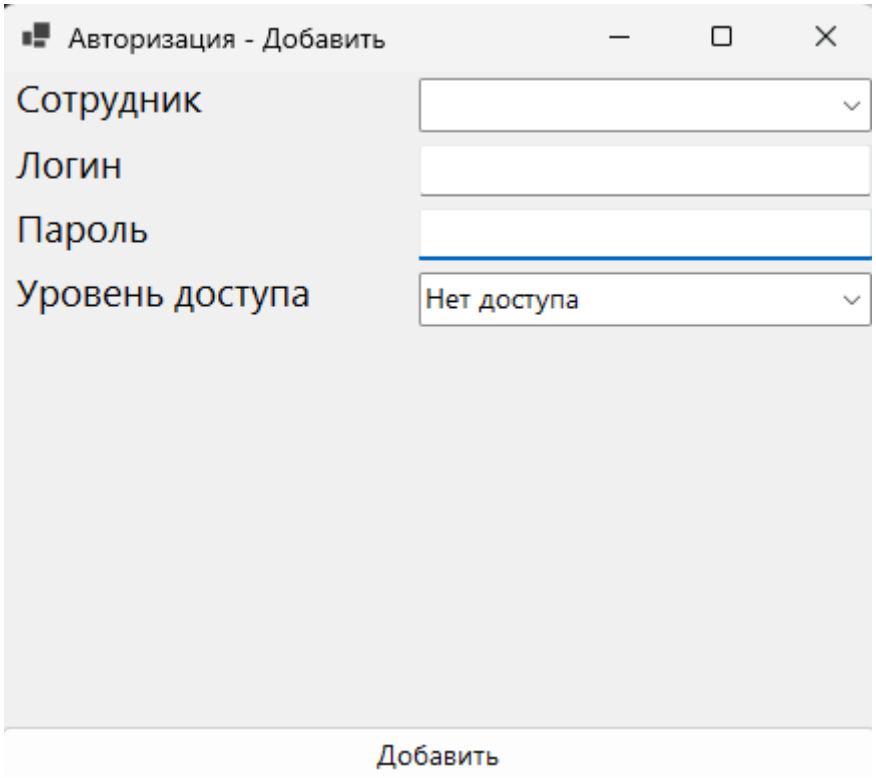

Рисунок 3.18. Окно добавления данных авторизации сотрудника

Клиенты

ID	Организация	Контактное лицо	Номер телефона	Электронная почта	Адрес
1	ООО "Солёный хлеб"	Анна Пирожкова	+7 (901) 111-11-11	anna@bread.ru	Москва, ул. Булочная, 12
2	ИП Васильев	Васильев Пётр	+7 (902) 222-22-22	vasilev@meat.ru	Раменское, ул. Мясная, 5
3	ООО "Морепродукт"	Светлана Рыбкина	+7 (903) 333-33-33	sveta@seafood.ru	Калининград, наб. Приморская, 7
4	ООО "Сладкоежка"	Николай Конфеткин	+7 (904) 444-44-44	kolya@sweet.ru	Тула, ул. Шоколадная, 3
5	ООО "Чистый дом"	Екатерина Моющая	+7 (905) 555-55-55	ekaterina@clean.ru	Люберцы, ул. Уборщиков, 10
6	ООО "Хлебозавод №1"	Михаил Тестов	+7 (906) 666-66-66	misha@dough.ru	Домодедово, пр-д Пекарский, 1
7	ООО "Мясокомбинат Восток"	Татьяна Говядина	+7 (907) 777-77-77	tanya@meat-east.ru	Балашиха, ул. Скотобойная, 15
8	ООО "Супермаркет Юг"	Роман Кассир	+7 (908) 888-88-88	roman@market-south.ru	Волгоград, пр-т Стачки, 50
9	ООО "Фудсервис"	Людмила Обедова	+7 (909) 999-99-99	luda@foodservice.ru	Самара, ул. Кухонная, 22
10	ООО "Кейтеринг Плюс"	Станислав Банкетов	+7 (910) 000-00-00	stas@catering.ru	Казань, ул. Праздничная, 8
11	ООО "Сеть Кафе"	Оксана Перекусова	+7 (911) 111-22-33	oksana@cafe.ru	Нижний Новгород, ул. Фастфуд, 12
12	ООО "Пицца Мания"	Дмитрий Тесто	+7 (912) 222-33-44	dima@pizza.ru	Ростов-на-Дону, ул. Дровяная, 5
13	ООО "Кондитерская №1"	Ирина Сахарова	+7 (913) 333-44-55	irina@cake.ru	Уфа, ул. Ванильная, 7
14	ООО "Сырный рай"	Владимир Брынза	+7 (914) 444-55-66	vlad@cheese.ru	Пермь, ул. Молочная, 3

Добавить

Изменить

Удалить

Обновить

Рисунок 3.19. Окно просмотр данных клиентов

Клиенты - Изменить

Организация: ООО "Солёный хлеб"

Контактное лицо: Анна Пирожкова

Номер телефона: +7 (901) 111-11-11

Электронная почта: anna@bread.ru

Адрес: Москва, ул. Булочная, 12

Изменить

Рисунок 3.20. Окно изменения данных клиента

Товары

ID	Поставщик	Наименование	Категория	Единица измерения	Цена за ед. товара	Описание
1	ООО РУССОЛЬ	Соль нитритная 0,6 комплексная 25кг	Пищевая добавка	мешок 25 кг	4200,00	Для производства мясопродуктов
2	ООО АгроСоль	Концентрат минеральный ГАЛИТ тип С 50кг	Минералы	мешок 50 кг	1800,00	Высший сорт, Илецк
3	ЗАО Илецкий карьер	Концентрат минеральный ГАЛИТ тип С 1т МКР	Минералы	МКР 1 т	35000,00	Тип С, Илецк
4	ЗАО Илецкий карьер	Концентрат минеральный ГАЛИТ высший сорт 1т	Минералы	МКР 1 т	37000,00	Высший сорт, месторождение Илецк
5	ООО ХимПродукт	Сода пищевая ГОСТ 2156-76 500г	Сода пищевая	пачка 500 г	120,00	Упаковка 24 шт., сорт 2
6	ООО ХимПродукт	Сода пищевая ГОСТ 2156-76 25кг	Сода пищевая	мешок 25 кг	3800,00	Полипропиленовый мешок
7	ООО Сладкая жизнь	Сахар песок ГОСТ 21-94	Сахар	мешок 50 кг	2200,00	Высший сорт
8	ООО РУССОЛЬ	Соль для животноводства брикетированная 10кг	Соль для животных	брикет 10 кг	600,00	Высший сорт
		Соль пищевая				С

Добавить

Изменить

Удалить

Обновить

Рисунок 3.21. Окно просмотр данных товаров

Товары - Изменить

Поставщик: ООО РУССОЛЬ

Наименование: Соль нитритная 0,6 комплексная 25кг

Категория: Пищевая добавка

Единица измерения: мешок 25 кг

Цена за ед. товара: 4200,00

Описание: Для производства мясопродуктов

Изменить

Рисунок 3.22. Окно изменения данных о товарах

Каждое окно добавления/изменения данных перед внесением изменений в БД проводит валидацию введённых данных. При не корректных данных окно выводит сообщение об ошибке ввода и указывает, где была совершена ошибка.

Для удобного выбора клиента, сотрудника или товара в окнах с изменение или добавлением предусмотрены выпадающие списки (ComboBox) данные в которые подгружаться в момент инициализации окна. Такое решение избавляет от ошибки связей в БД и гарантирует что ID выбранного объекта существует в базе и не вызовет критической ошибки.

3.3. Проектирование структуры базы данных

База данных (БД) — это организованная и структурированная коллекция данных, хранящаяся в электронном виде на компьютере или сервере. Она предназначена для удобного и эффективного хранения, поиска, управления и обработки больших объёмов информации. Вместо того чтобы хранить данные в отдельных файлах или бумажных анкетах, БД объединяет их в единую систему, где они связаны между собой определённым образом.

Основные цели базы данных:

1) Эффективное хранение информации: обеспечить структурированное и надёжное хранение больших объёмов данных.

2) Управление данными: предоставить средства для добавления, изменения, удаления и поиска данных.

3) Обеспечение доступа: обеспечить пользователям и приложениям быстрый и контролируемый доступ к нужной информации.

4) Поддержание целостности и безопасности: обеспечить точность, согласованность и защиту данных от несанкционированного доступа или повреждений.

5) Снижение избыточности: минимизировать дублирование данных, храня информацию в одном месте и ссылаясь на неё из других частей системы.

Основные элементы базы данных:

1) Таблицы (Tables): Основной элемент БД. Представляет собой структуру, похожую на электронную таблицу, состоящую из строк и столбцов. Каждая таблица хранит информацию об одном типе сущности (например, "Клиенты", "Заказы", "Товары").

2) Поля (Fields / Columns): Столбцы таблицы. Определяют какие атрибуты или характеристики хранятся для каждой записи (например, "Имя клиента", "Дата заказа", "Цена").

3) Записи (Records / Rows): Строки таблицы. Каждая запись представляет собой один экземпляр сущности, содержащий конкретные значения для всех полей (например, одна строка в таблице "Клиенты" — это информация об одном конкретном клиенте).

4) Ключи (Keys):

- Первичный ключ (Primary Key): Особое поле (или комбинация полей), которое уникальным образом идентифицирует каждую запись в таблице. Обычно это числовой идентификатор.
- Внешний ключ (Foreign Key): Поле (или набор полей), которое используется для установления связи между двумя таблицами. Оно ссылается на первичный ключ в другой (или той же) таблице.

Система управления базами данных (СУБД) — это специализированное программное обеспечение, которое предоставляет интерфейс между базой данных, пользователями и приложениями. СУБД отвечает за создание, управление, обслуживание и использование базы данных. Она обеспечивает выполнение запросов к БД (например, SQL-запросов), поддерживает целостность данных, управляет параллельным доступом, обеспечивает безопасность и позволяет администрировать базу данных.

Примеры СУБД: Microsoft SQL Server, MySQL, PostgreSQL, Oracle Database, SQLite. В рамках дипломного проекта для информационной системы ООО «Руссоль-центр» используется Microsoft SQL Server как надёжная и мощная реляционная СУБД, хорошо интегрирующаяся с выбранной платформой разработки (.NET и C#).

3.3.1. Физическая ER диаграмма

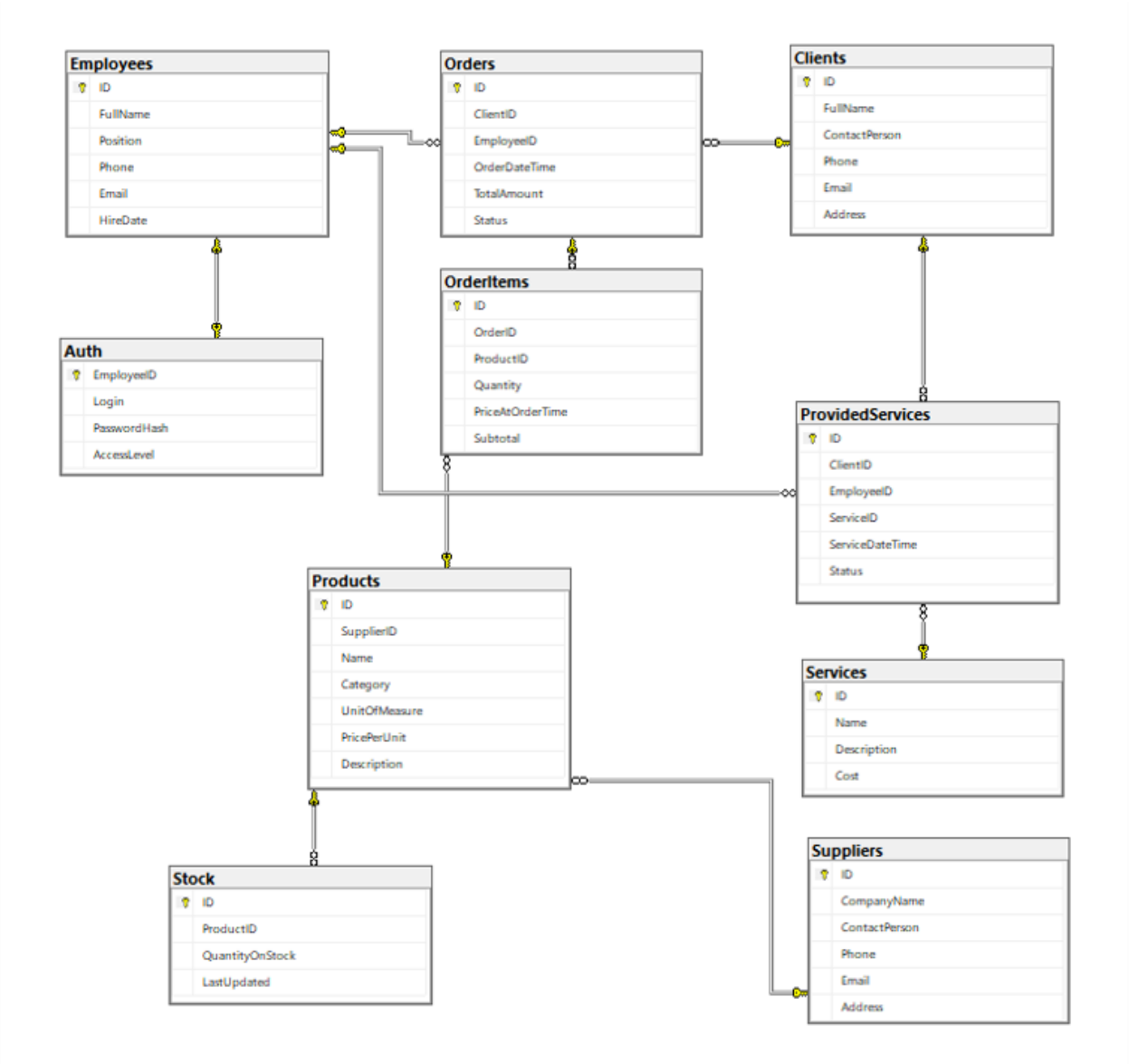

Рисунок 3.23. Физическая ER диаграмма

3.3.2. Описание таблиц

База данных имеет несколько таблиц каждая из которых играет важную роль.

Описание таблиц:

1) Поставщики (Suppliers)

- ID (ID) — уникальный идентификатор поставщика (INT, IDENTITY, PRIMARY KEY, NOT NULL)
- CompanyName (Наименование компании) — полное наименование компании-поставщика (NVARCHAR(255), NOT NULL)

• ContactPerson (Контактное лицо) — ФИО контактного лица в компании-поставщике (NVARCHAR(255), NULLABLE)

• Phone (Телефон) — контактный номер телефона поставщика (NVARCHAR(50), NULLABLE)

• Email (Email) — адрес электронной почты поставщика (NVARCHAR(255), NULLABLE)

• Address (Адрес) — юридический или фактический адрес компании-поставщика (NVARCHAR(500), NULLABLE)

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	☐
	CompanyName	nvarchar(255)	☐
	ContactPerson	nvarchar(255)	☑
	Phone	nvarchar(50)	☑
	Email	nvarchar(255)	☑
	Address	nvarchar(500)	☑

Рисунок 3.24. Таблица «Suppliers»

	ID	CompanyName	ContactPerson	Phone	Email	Address
1	1	ООО РУССОЛЬ	Иван Петров	+7 (495) 123-45-67	info@russol.ru	г. Новомосковск, ул. Соляная, 1
2	2	ЗАО Илецкий карьер	Марина Сидорова	+7 (345) 234-56-78	zakaz@ileck.ru	г. Илецк, шахта №3
3	3	ООО Мясной мир	Алексей Кузнецов	+7 (499) 345-67-89	supply@myasnoy.ru	Москва, пр-д Колбасный, 5
4	4	ООО Сладкая жизнь	Елена Федорова	+7 (812) 456-78-90	sugar@sladko.ru	Санкт-Петербург, ул. Конфетная, 10
5	5	ООО ХимПродукт	Дмитрий Морозов	+7 (411) 567-89-01	chem@soda.ru	Новосибирск, химзавод, 2
6	6	ООО Солерудник	Ольга Волкова	+7 (495) 222-33-44	geolog@solerud.ru	Тульская область, п. Галитный
7	7	ООО Белый кристалл	Сергей Попов	+7 (845) 333-44-55	sales@whitecrystal.ru	Саратов, ул. Промышленная, 12
8	8	ООО ТоргСоль	Наталья Крылова	+7 (473) 444-55-66	trade@torgsol.ru	Воронеж, ул. Складская, 8
9	9	ООО Южсолепродукт	Андрей Михайлов	+7 (861) 555-66-77	south@saltprod.ru	Краснодар, шоссе Нефтяников, 15
10	10	ООО УралХимСоль	Евгений Орлов	+7 (343) 666-77-88	ural@khim-sol.ru	Екатеринбург, пр-кт Ленина, 40
11	11	ООО Северсоль	Людмила Нестерова	+7 (818) 777-88-99	north@seversol.ru	Архангельск, ул. Полярная, 5
12	12	ООО АгроСоль	Виктор Титов	+7 (846) 888-99-00	agro@agrosol.ru	Самара, с. Солёное
13	13	ООО КавказСоль	Руслан Магомедов	+7 (878) 999-00-11	kavkaz@kavsol.ru	Пятигорск, ул. Горная, 3
14	14	ООО Сибсолепром	Ирина Широкова	+7 (383) 333-44-55	sib@prom-sol.ru	Новосибирск, ул. Заводская, 10
15	15	ООО Чистая соль	Денис Ковалёв	+7 (424) 444-55-66	clean@chistsol.ru	Южно-Сахалинск, пр-т Мира, 22
16	16	ООО МореСоль	Алёна Васильева	+7 (421) 555-66-77	more@mor-esol.ru	Хабаровск, наб. Речная, 7
17	17	ООО Эко-Соль	Павел Зайцев	+7 (865) 666-77-88	eco@ekosol.ru	Ставрополь, ул. Природная, 14
18	18	ООО Соляной двор	Тамара Петрова	+7 (495) 777-88-99	dvor@sol-dvor.ru	Москва, ул. Крестьянская, 1
19	19	ООО Солёный берег	Анатолий Лебедев	+7 (846) 888-99-00	beret@solbere.ru	Самара, наб. Волги, 100
20	20	ООО Солеперерабо...	Максим Гришин	+7 (495) 999-00-11	process@solproc.ru	Подольск, ул. Техническая, 33

Рисунок 3.25. Данные из таблицы «Suppliers»

2) Товары (Products)

• ID (ID) — уникальный идентификатор товара (INT, IDENTITY, PRIMARY KEY, NOT NULL)

- SupplierID (ID Поставщика) — внешний ключ, ссылающийся на таблицу Suppliers (INT, NOT NULL, FOREIGN KEY -> Suppliers.ID)
- Name (Название) — наименование товара (NVARCHAR(255), NOT NULL)
- Category (Категория) — категория, к которой относится товар (NVARCHAR(100), NULLABLE)
- UnitOfMeasure (Единица измерения) — единица измерения товара (например, кг, шт) (NVARCHAR(50), NULLABLE)
- PricePerUnit (Цена за единицу) — цена одной единицы товара (DECIMAL(10,2), NOT NULL, CHECK >= 0)
- Description (Описание) — подробное описание товара (NVARCHAR(MAX), NULLABLE)

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	☐
	SupplierID	int	☐
	Name	nvarchar(255)	☐
	Category	nvarchar(100)	☑
	UnitOfMeasure	nvarchar(50)	☑
	PricePerUnit	decimal(10, 2)	☐
	Description	nvarchar(MAX)	☑

Рисунок 3.26. Таблица «Products»

	ID	SupplierID	Name	Category	UnitOfMeasure	PricePerUnit	Description
1	1	1	Соль нитритная 0,6 комплексная 25кг	Пищевая добавка	мешок 25 кг	4200.00	Для производства мясопродуктов
2	2	12	Концентрат минеральный ГАЛИТ тип С 50кг	Минералы	мешок 50 кг	1800.00	Высший сорт, Илецк
3	3	2	Концентрат минеральный ГАЛИТ тип С 1т МКР	Минералы	МКР 1 т	35000.00	Тип С, Илецк
4	4	2	Концентрат минеральный ГАЛИТ высший сорт 1т	Минералы	МКР 1 т	37000.00	Высший сорт, месторождение Илецк
5	5	5	Сода пищевая ГОСТ 2156-76 500г	Сода пищевая	пачка 500 г	120.00	Упаковка 24 шт., сорт 2
6	6	5	Сода пищевая ГОСТ 2156-76 25кг	Сода пищевая	мешок 25 кг	3800.00	Полипропиленовый мешок
7	7	4	Сахар песок ГОСТ 21-94	Сахар	мешок 50 кг	2200.00	Высший сорт
8	8	1	Соль для животноводства брикетированная 10кг	Соль для животных	брикет 10 кг	600.00	Высший сорт
9	9	1	Соль пищевая выварочная экстра 25кг	Пищевая соль	мешок 25 кг	1200.00	С противослеживающей добавкой, Новомосковск
10	10	1	Соль пищевая выварочная экстра 50кг	Пищевая соль	мешок 50 кг	2300.00	С противослеживающей добавкой, Новомосковск
11	11	1	Соль пищевая выварочная экстра 1т	Пищевая соль	МКР 1 т	45000.00	Новомосковск
12	12	1	Соль пищевая выварочная экстра 1кг бумажный пакет	Пищевая соль	пакет 1 кг	260.00	Упаковка 10 шт., Новомосковск
13	13	1	Соль пищевая выварочная экстра 0,25кг п/п туба	Пищевая соль	туба 250 г	75.00	Термоусадочная плёнка, 24 шт.
14	14	1	Соль пищевая выварочная экстра 0,5кг п/п туба	Пищевая соль	туба 500 г	140.00	Термоусадочная плёнка
15	15	2	Соль пищевая мелкая Илецкая В/С 180г	Пищевая соль	пачка 180 г	80.00	Месторождение Илецк
16	16	2	Соль пищевая мелкая Илецкая В/С 500г	Пищевая соль	пачка 500 г	190.00	Месторождение Илецк
17	17	2	Соль пищевая мелкая Просто Азбука В/С 180г	Пищевая соль	пачка 180 г	75.00	Эконом-линейка
18	18	2	Соль пищевая мелкая Просто Азбука В/С 500г	Пищевая соль	пачка 500 г	185.00	Эконом-линейка
19	19	1	Соль пищевая молотая морская В/С 600г	Пищевая соль	картон 600 г	320.00	Помол №3, упаковка 24 шт.
20	20	1	Соль пищевая молотая В/С 1кг картон	Пищевая соль	пачка 1 кг	250.00	Помол №1, термоплёнка 21 шт.
21	21	1	Соль поваренная Экстра фасовка 1кг	Пищевая соль	пакет 1 кг	240.00	20 шт. в упаковке, РУССОЛЬ
22	22	1	Соль поваренная йодированная 1кг	Пищевая соль	пакет 1 кг	180.00	Помол №1, мешок 30 кг
23	23	1	Соль поваренная таблетированная 25кг	Пищевая соль	мешок 25 кг	2800.00	Таблетки 10-15г, Новомосковск
24	24	2	Соль таблетированная выварочная Экстра 25кг	Пищевая соль	мешок 25 кг	2750.00	Аналог РУССОЛЬ
25	25	1	Фасовка п/э 1кг помол №1	Фасовка	пакет 1 кг	15.00	50 шт. в мешке
26	26	1	Фасовка п/э 1кг высший сорт	Фасовка	пакет 1 кг	16.00	50 шт. в мешке
27	27	1	Сахар фасованный 1кг	Сахар	пакет 1 кг	70.00	ГОСТ 21-94, 20 шт. в коробке
28	28	5	Сода пищевая фасованная 250г	Сода пищевая	туба 250 г	65.00	Термоусадочная плёнка
29	29	5	Сода пищевая фасованная 500г	Сода пищевая	туба 500 г	110.00	Термоусадочная плёнка
30	30	2	Соль Илецкая Просто Азбука 180г ВкусВилл	Пищевая соль	пачка 180 г	85.00	Специальная упаковка для сети
31	31	1	Соль поваренная Экстра таблетированная 25кг	Пищевая соль	мешок 25 кг	2800.00	Таблетки 10-15г, аналог РУССОЛЬ

Рисунок 3.27. Данные из таблицы «Products»

3) Склад (Stock)

- ID (ID) — уникальный идентификатор записи на складе (INT, IDENTITY, PRIMARY KEY, NOT NULL)
- ProductID (ID Товара) — внешний ключ, ссылающийся на таблицу Products (INT, NOT NULL, FOREIGN KEY -> Products.ID)
- QuantityOnStock (Количество на складе) — текущее количество товара в наличии (DECIMAL(12,2), NOT NULL, CHECK >= 0)
- LastUpdated (Последнее обновление) — дата и время последнего изменения остатка (DATETIME2, DEFAULT GETDATE())

Имя столбца	Тип данных	Разрешить знач...
ID	int	☐
ProductID	int	☐
QuantityOnStock	decimal(12, 2)	☐
LastUpdated	datetime2(7)	☑

Рисунок 3.28. Таблица «Stock»

	ID	ProductID	QuantityOnStock	LastUpdated
1	1	1	170.01	2025-10-26 21:48:35.7600000
2	2	2	80.00	2025-10-26 21:48:35.7633333
3	3	3	7.00	2025-10-26 21:48:35.7633333
4	4	4	53.00	2025-10-26 21:48:35.7700000
5	5	5	300.00	2025-10-26 21:48:35.7700000
6	6	6	40.00	2025-10-26 21:48:35.7700000
7	7	7	60.00	2025-10-26 21:48:35.7700000
8	8	8	75.00	2025-10-26 21:48:35.7733333
9	9	9	150.00	2025-10-26 21:48:35.7733333
10	10	10	80.00	2025-10-26 21:48:35.7733333
11	11	11	5.00	2025-10-26 21:48:35.7733333
12	12	12	100.00	2025-10-26 21:30:20.6066667
13	13	13	200.00	2025-10-26 21:30:20.6100000
14	14	14	180.00	2025-10-26 21:30:20.6100000
15	15	15	200.00	2025-10-26 21:30:20.6133333
16	16	16	120.00	2025-10-26 21:30:20.6133333
17	17	17	100.00	2025-10-26 21:30:20.6133333
18	18	18	100.00	2025-10-26 21:30:20.6166667
19	19	19	80.00	2025-10-26 21:48:35.7766667
20	20	20	90.00	2025-10-26 21:48:35.7766667
21	21	21	200.00	2025-10-26 21:30:20.6200000
22	22	22	150.00	2025-10-26 21:30:20.6233333
23	23	23	95.00	2025-10-26 21:30:20.6233333
24	24	24	70.00	2025-10-26 21:48:35.7766667
25	25	25	88.00	2025-10-26 21:30:20.6266667
26	26	26	50.00	2025-10-26 21:30:20.6266667
27	27	27	30.00	2025-10-26 21:30:20.6300000
28	28	28	250.00	2025-10-26 21:48:35.7766667
29	29	29	180.00	2025-10-26 21:30:20.6300000
30	30	30	70.00	2025-10-26 21:30:20.6333333
31	31	31	85.00	2025-10-26 21:30:20.6333333

Рисунок 3.29. Данные из таблицы «Stock»

4) Клиенты (Clients)

- ID (ID) — уникальный идентификатор клиента (INT, IDENTITY, PRIMARY KEY, NOT NULL)
- FullName (Полное имя) — наименование компании или ФИО клиента (NVARCHAR(255), NOT NULL)
- ContactPerson (Контактное лицо) — ФИО контактного лица у клиента (NVARCHAR(255), NULLABLE)
- Phone (Телефон) — контактный номер телефона клиента (NVARCHAR(50), NULLABLE)

- Email (Email) — адрес электронной почты клиента (NVARCHAR(255), NULLABLE)
- Address (Адрес) — адрес клиента (NVARCHAR(500), NULLABLE)

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	☐
	FullName	nvarchar(255)	☐
	ContactPerson	nvarchar(255)	☑
	Phone	nvarchar(50)	☑
	Email	nvarchar(255)	☑
	Address	nvarchar(500)	☑

Рисунок 3.30. Таблица «Clients»

	ID	FullName	ContactPerson	Phone	Email	Address
1	1	ООО "Солёный хлеб"	Анна Пирожкова	+7 (901) 111-11-11	anna@bread.ru	Москва, ул. Булочная, 12
2	2	ИП Васильев	Васильев Пётр	+7 (902) 222-22-22	vasilev@meat.ru	Раменское, ул. Мясная, 5
3	3	ООО "Морепродукт"	Светлана Рыбкина	+7 (903) 333-33-33	sveta@seafood.ru	Калининград, наб. Приморская, 7
4	4	ООО "Сладкоежка"	Николай Конфеткин	+7 (904) 444-44-44	kolya@sweet.ru	Тула, ул. Шоколадная, 3
5	5	ООО "Чистый дом"	Екатерина Моющая	+7 (905) 555-55-55	ekaterina@clean.ru	Люберцы, ул. Уборщиков, 10
6	6	ООО "Хлебозавод №1"	Михаил Тестов	+7 (906) 666-66-66	misha@dough.ru	Домодедово, пр-д Пекарский, 1
7	7	ООО "Мясокомбинат Восток"	Татьяна Говядина	+7 (907) 777-77-77	tanya@meat-east.ru	Балашиха, ул. Скотобойная, 15
8	8	ООО "Супермаркет Юг"	Роман Кассир	+7 (908) 888-88-88	roman@market-south.ru	Волгоград, пр-т Стачки, 50
9	9	ООО "Фудсервис"	Людмила Обедова	+7 (909) 999-99-99	luda@foodservice.ru	Самара, ул. Кухонная, 22
10	10	ООО "Кейтеринг Плюс"	Станислав Банкетов	+7 (910) 000-00-00	stas@catering.ru	Казань, ул. Праздничная, 8
11	11	ООО "Сеть Кафе"	Оксана Перекусова	+7 (911) 111-22-33	oksana@cafe.ru	Нижний Новгород, ул. Фастфуд, 12
12	12	ООО "Пицца Мания"	Дмитрий Тесто	+7 (912) 222-33-44	dima@pizza.ru	Ростов-на-Дону, ул. Дровяная, 5
13	13	ООО "Кондитерская №1"	Ирина Сахарова	+7 (913) 333-44-55	irina@cake.ru	Уфа, ул. Ванильная, 7
14	14	ООО "Сырный рай"	Владимир Брынза	+7 (914) 444-55-66	vlad@cheese.ru	Пермь, ул. Молочная, 3
15	15	ООО "Колбасный цех"	Геннадий Фарш	+7 (915) 555-66-77	gena@sausage.ru	Челябинск, ул. Коптильная, 9
16	16	ООО "Рыбный рынок"	Нина Камбала	+7 (916) 666-77-88	nina@fish.ru	Мурманск, пр-т Морской, 11
17	17	ООО "Фермерский двор"	Борис Коровкин	+7 (917) 777-88-99	boris@ferma.ru	Тверь, д. Село, 1
18	18	ООО "Бакалея Центр"	Любовь Бакалейкина	+7 (918) 888-99-00	lyuba@bakeley.ru	Ярославль, ул. Торговая, 15
19	19	ООО "Снэк-Мастер"	Артём Чипсов	+7 (919) 999-00-11	artem@snack.ru	Омск, ул. Солёная, 2
20	20	ООО "Лаборатория Вкуса"	Профессор Специи	+7 (920) 000-11-22	prof@vkus.ru	Красноярск, научный городок, 1
21	21	ООО "Гастроном №1"	Татьяна Шефова	+7 (921) 111-22-33	gastron@shop.ru	Владивосток, ул. Кулинарная, 10
22	22	ООО "Молоко и соль"	Олег Сливкин	+7 (922) 222-33-44	molo@solmoloko.ru	Курск, ул. Молочная, 8
23	23	ООО "Солёные огурцы"	Валентина Рассолк...	+7 (923) 333-44-55	valya@pickle.ru	Тамбов, ул. Консервная, 12
24	24	ООО "Суповой набор"	Игорь Бульонов	+7 (924) 444-55-66	igor@soup.ru	Иркутск, ул. Кухня, 1
25	25	ООО "Шашлычная на углу"	Рустам Мангалов	+7 (925) 555-66-77	rustam@shashlik.ru	Казань, ул. Барбекю, 3

Рисунок 3.31. Данные из таблицы «Clients»

5) Сотрудники (Employees)

- ID (ID) — уникальный идентификатор сотрудника (INT, IDENTITY, PRIMARY KEY, NOT NULL)
- FullName (Полное имя) — ФИО сотрудника (NVARCHAR(255), NOT NULL)
- Position (Должность) — занимаемая должность (NVARCHAR(100), NULLABLE)

• Phone (Телефон) — контактный номер телефона сотрудника (NVARCHAR(50), NULLABLE)

• Email (Email) — адрес электронной почты сотрудника (NVARCHAR(255), NULLABLE)

• HireDate (Дата найма) — дата, когда сотрудник был принят на работу (DATE, NOT NULL, DEFAULT GETDATE())

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	☐
	FullName	nvarchar(255)	☐
	Position	nvarchar(100)	☑
	Phone	nvarchar(50)	☑
	Email	nvarchar(255)	☑
	HireDate	date	☐

Рисунок 3.32. Таблица «Employees»

	ID	FullName	Position	Phone	Email	HireDate
1	1	Козлов Артем Михайлович	Менеджер по продажам	+7 (900) 111-22-33	kozlov@saltwholesale.ru	2023-01-10
2	2	Волкова Елена Сергеевна	Бухгалтер	+7 (900) 222-33-44	volkova@saltwholesale.ru	2023-02-15
3	3	Смирнов Дмитрий Андреевич	Складской работник	+7 (900) 333-44-55	smirnov@saltwholesale.ru	2023-03-01
4	4	Петрова Ольга Николаевна	Администратор	+7 (900) 444-55-66	petrova@saltwholesale.ru	2023-04-05
5	5	Иванов Максим Львович	Технический специалист	+7 (900) 555-66-77	ivanov@saltwholesale.ru	2023-05-12
6	6	11111	1111	1111	111	2025-11-13

Рисунок 3.33. Данные из таблицы «Employees»

6) Авторизация (Auth)

• EmployeeID (ID Сотрудника) — внешний ключ, ссылающийся на таблицу Employees и одновременно являющийся первичным ключом (INT, PRIMARY KEY, NOT NULL, FOREIGN KEY -> Employees.ID)

• Login (Логин) — уникальное имя пользователя для входа в систему (NVARCHAR(100), UNIQUE, NOT NULL)

• PasswordHash (Хэш пароля) — зашифрованное значение пароля (NVARCHAR(255), NOT NULL)

• AccessLevel (Уровень доступа) — числовое значение, определяющее роль и права доступа (INT, NOT NULL, DEFAULT 0, CHECK >= -1). Значение -1 означает заблокированного пользователя.

	Имя столбца	Тип данных	Разрешить знач...
🔑	EmployeeID	int	☐
	Login	nvarchar(100)	☐
	PasswordHash	nvarchar(255)	☐
	AccessLevel	int	☐

Рисунок 3.34. Таблица «Auth»

	EmployeeID	Login	PasswordHash	AccessLevel
1	1	kozlov	123456	0
2	2	volkova	123456	1
3	3	smirnov	123456	2
4	4	petrova	123456	-1
5	5	ivanov	1234	0
6	6	1111	1111	1

Рисунок 3.35. Данные из таблицы «Auth»

7) Заказы (Orders)

- ID (ID) — уникальный идентификатор заказа (INT, IDENTITY, PRIMARY KEY, NOT NULL)
- ClientID (ID Клиента) — внешний ключ, ссылающийся на таблицу Clients (INT, NOT NULL, FOREIGN KEY -> Clients.ID)
- EmployeeID (ID Сотрудника) — внешний ключ, ссылающийся на таблицу Employees (INT, NOT NULL, FOREIGN KEY -> Employees.ID)
- OrderDateTime (Дата и время заказа) — дата и время создания заказа (DATETIME2, NOT NULL, DEFAULT GETDATE())
- TotalAmount (Общая сумма) — общая сумма заказа (DECIMAL(12,2), NOT NULL, CHECK >= 0)
- Status (Статус) — текущий статус заказа (например, "Новый", "В обработке") (NVARCHAR(50), NOT NULL, DEFAULT 'Новый')

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	☐
	ClientID	int	☐
	EmployeeID	int	☐
	OrderDateTime	datetime2(7)	☐
	TotalAmount	decimal(12, 2)	☐
	Status	nvarchar(50)	☐

Рисунок 3.36. Таблица «Orders»

	ID	ClientID	EmployeeID	OrderDateTime	TotalAmount	Status
1	2	2	1	2025-01-12 14:20:00.0000000	18500.00	Завершён
2	3	3	2	2025-01-15 09:15:00.0000000	52000.00	Завершён
3	4	4	1	2025-01-18 16:45:00.0000000	4080.00	В процессе
4	5	5	3	2025-01-20 11:00:00.0000000	32000.00	В процессе
5	6	6	2	2025-01-22 13:30:00.0000000	15000.00	Создан
6	7	7	1	2025-01-25 15:10:00.0000000	78000.00	Создан
7	8	8	4	2025-01-27 10:05:00.0000000	12500.00	Создан
8	9	9	5	2025-01-28 17:20:00.0000000	67000.00	Новый
9	10	10	3	2025-01-29 12:40:00.0000000	43000.00	Новый
10	11	11	1	2025-02-01 09:10:00.0000000	22000.00	Отменён
11	12	12	2	2025-02-02 11:25:00.0000000	31000.00	Завершён
12	13	13	3	2025-02-03 14:40:00.0000000	17500.00	В процессе
13	14	14	1	2025-02-04 16:15:00.0000000	8800.00	Создан
14	15	15	4	2025-02-05 10:50:00.0000000	41000.00	Новый
15	16	16	5	2025-02-06 13:20:00.0000000	19500.00	Новый
16	17	17	2	2025-02-07 15:35:00.0000000	54000.00	Создан
17	18	18	3	2025-02-08 09:45:00.0000000	28000.00	В процессе
18	19	19	1	2025-02-09 12:00:00.0000000	36000.00	Завершён
19	20	20	4	2025-02-10 14:10:00.0000000	13200.00	Создан
20	21	21	5	2025-02-11 16:30:00.0000000	9700.00	Новый
21	22	22	1	2025-02-12 10:20:00.0000000	44000.00	Создан
22	23	23	2	2025-02-13 11:30:00.0000000	18000.00	Новый
23	24	24	3	2025-02-14 12:40:00.0000000	29500.00	Создан
24	25	25	1	2025-02-15 13:50:00.0000000	33000.00	В процессе

Рисунок 3.37. Данные из таблицы «Orders»

8) Товары в заказе (OrderItems)

- ID (ID) — уникальный идентификатор позиции в заказе (INT, IDENTITY, PRIMARY KEY, NOT NULL)
- OrderID (ID Заказа) — внешний ключ, ссылающийся на таблицу Orders (INT, NOT NULL, FOREIGN KEY -> Orders.ID)
- ProductID (ID Товара) — внешний ключ, ссылающийся на таблицу Products (INT, NOT NULL, FOREIGN KEY -> Products.ID)
- Quantity (Количество) — количество единиц товара в позиции заказа (DECIMAL(12,2), NOT NULL, CHECK > 0)
- PriceAtOrderTime (Цена на момент заказа) — цена единицы товара на момент оформления заказа (DECIMAL(10,2), NOT NULL, CHECK >= 0)

- Subtotal (Итого) — вычисляемое поле: Quantity * PriceAtOrderTime (PERSISTED, DECIMAL(22,4))

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	☐
	OrderID	int	☐
	ProductID	int	☐
	Quantity	decimal(12, 2)	☐
	PriceAtOrderTime	decimal(10, 2)	☐
	Subtotal		☑

Рисунок 3.38. Таблица «OrderItems»

	ID	OrderID	ProductID	Quantity	PriceAtOrderTime	Subtotal
1	3	2	5	100.00	190.00	19000.0000
2	4	2	8	5.00	600.00	3000.0000
3	5	3	3	1.00	45000.00	45000.0000
4	6	3	6	2.00	3800.00	7600.0000
5	9	5	6	10.00	2800.00	28000.0000
6	10	5	11	10.00	4200.00	42000.0000
7	11	6	7	1.00	2200.00	2200.0000
8	12	6	19	5.00	320.00	1600.0000
9	13	7	13	1.00	35000.00	35000.0000
10	14	7	14	100.00	320.00	32000.0000
11	15	8	20	100.00	65.00	6500.0000
12	16	8	21	100.00	110.00	11000.0000
13	17	9	16	200.00	180.00	36000.0000
14	18	9	22	100.00	75.00	7500.0000
15	19	10	18	5.00	2750.00	13750.0000
16	20	10	23	100.00	185.00	18500.0000
17	21	12	9	20.00	1200.00	24000.0000
18	22	12	15	50.00	190.00	9500.0000
19	23	13	10	100.00	2300.00	230000.0...
20	24	13	30	5.00	600.00	3000.0000
21	25	14	17	60.00	15.00	900.0000
22	26	14	27	20.00	70.00	1400.0000
23	27	15	24	150.00	250.00	37500.0000
24	28	15	29	200.00	110.00	22000.0000
25	29	16	1	5.00	1200.00	6000.0000
26	30	16	26	10.00	180.00	1800.0000
27	31	17	3	1.00	45000.00	45000.0000
28	32	17	14	30.00	320.00	9600.0000
29	33	18	6	10.00	2800.00	28000.0000
30	34	18	11	5.00	4200.00	21000.0000
31	35	19	25	200.00	70.00	14000.0000
32	36	19	28	100.00	65.00	6500.0000
33	37	20	21	120.00	110.00	13200.0000
34	38	20	19	100.00	320.00	32000.0000
35	39	21	5	100.00	120.00	12000.0000
36	40	21	30	10.00	600.00	6000.0000
37	41	22	13	1.00	35000.00	35000.0000
38	42	22	14	30.00	320.00	9600.0000

Рисунок 3.39. Данные из таблицы «OrderItems»

9) Услуги (Services)

- ID (ID) — уникальный идентификатор услуги (INT, IDENTITY, PRIMARY KEY, NOT NULL)
- Name (Название) — наименование услуги (NVARCHAR(255), NOT NULL)
- Description (Описание) — подробное описание услуги (NVARCHAR(MAX), NULLABLE)
- Cost (Стоимость) — стоимость оказания услуги (DECIMAL(10,2), NOT NULL, CHECK >= 0)

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	☐
	Name	nvarchar(255)	☐
	Description	nvarchar(MAX)	☑
	Cost	decimal(10, 2)	☐

Рисунок 3.40. Таблица «Services»

	ID	Name	Description	Cost
1	1	Доставка до склада клиента	Транспортировка заказа до указанного адреса	5000.00
2	2	Консультация по подбору соли	Профессиональная консультация технолога	2000.00
3	3	Подготовка сертификатов качества	Оформление документов на продукцию	1500.00
4	4	Гибкая упаковка под заказ	Фасовка продукции по индивидуальным мешкам	3000.00
5	5	Аналитика состава	Лабораторный анализ образца соли	4000.00
6	6	Обучение персонала	Инструктаж по работе с солью и добавками	2500.00
7	7	Разработка рецептуры	Создание формулы с использованием соли	3500.00
8	8	Контроль качества при получении	Проверка партии на соответствие ГОСТ	1800.00
9	9	Упаковка в брикеты	Прессование соли в брикеты 10 кг	1200.00
10	10	Фасовка в п/э пакеты	Разделение на упаковки по 1 кг	800.00
11	11	Маркировка продукции	Нанесение этикеток согласно ТР ТС	600.00
12	12	Создание онлайн-каталога	Размещение товаров в цифровом виде	7000.00
13	13	Поддержка API интеграции	Настройка обмена данными с ERP системами	10000.00
14	14	Резервирование партии	Бронирование товара на складе	500.00
15	15	Экспресс-доставка	Доставка за 24 часа	8000.00
16	16	Возврат и утилизация	Приём некондиционного товара	2000.00
17	17	Организация выставки	Участие в отраслевых мероприятиях	15000.00
18	18	Техническое обслуживание оборудования	Обслуживание дозаторов и смесителей	4500.00
19	19	Консультация по логистике	Оптимизация маршрутов доставки	3000.00
20	20	Подбор аналогов	Замена товара при отсутствии на складе	1000.00
21	21	Консультация по фасовке	Подбор оптимального типа упаковки	1700.00
22	22	Документальное сопровождение	Оформление накладных, актов	1300.00
23	23	Предоставление пробников	Выдача бесплатных образцов	700.00
24	24	Оценка потребностей	Анализ объёмов потребления клиента	1900.00
25	25	Тренинг по безопасности	Обучение обращению с химическими добавками	2100.00
26	26	Организация контрольного замера	Проверка точности дозирования	2300.00

Рисунок 3.41. Данные из таблицы «Services»

10) Оказанные услуги (ProvidedServices)

- ID (ID) — уникальный идентификатор записи об оказанной услуге (INT, IDENTITY, PRIMARY KEY, NOT NULL)
- ClientID (ID Клиента) — внешний ключ, ссылающийся на таблицу Clients (INT, NOT NULL, FOREIGN KEY -> Clients.ID)
- EmployeeID (ID Сотрудника) — внешний ключ, ссылающийся на таблицу Employees (INT, NOT NULL, FOREIGN KEY -> Employees.ID)
- ServiceID (ID Услуги) — внешний ключ, ссылающийся на таблицу Services (INT, NOT NULL, FOREIGN KEY -> Services.ID)
- ServiceDateTime (Дата и время оказания) — дата и время, когда услуга была оказана (DATETIME2, NOT NULL, DEFAULT GETDATE())
- Status (Статус) — текущий статус оказания услуги (например, "В процессе", "Завершена") (NVARCHAR(50), NOT NULL, DEFAULT 'В процессе')

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	☐
	ClientID	int	☐
	EmployeeID	int	☐
	ServiceID	int	☐
	ServiceDateTime	datetime2(7)	☐
	Status	nvarchar(50)	☐

Рисунок 3.42. Таблица «ProvidedServices»

	ID	ClientID	EmployeeID	ServiceID	ServiceDateTime	Status
1	1	1	1	1	2025-01-11 11:00:00.0000000	Завершён
2	2	2	2	2	2025-01-13 15:30:00.0000000	Завершён
3	3	3	2	19	2025-01-16 10:00:00.0000000	Завершён
4	4	4	1	4	2025-01-19 14:00:00.0000000	В процессе
5	5	5	4	5	2025-01-21 12:00:00.0000000	Создан
6	6	6	2	1	2025-01-23 09:30:00.0000000	Завершён
7	7	7	1	2	2025-01-26 11:15:00.0000000	Завершён
8	8	8	5	3	2025-01-28 16:00:00.0000000	Создан
9	9	9	3	4	2025-01-29 10:00:00.0000000	Создан
10	10	10	1	5	2025-01-30 13:00:00.0000000	Новый
11	11	11	2	6	2025-02-01 10:00:00.0000000	Новый
12	12	12	3	7	2025-02-02 11:00:00.0000000	В процессе
13	13	13	1	8	2025-02-03 12:00:00.0000000	Создан
14	14	14	4	9	2025-02-04 13:00:00.0000000	Завершён
15	15	15	5	10	2025-02-05 14:00:00.0000000	В процессе
16	16	16	1	11	2025-02-06 15:00:00.0000000	Создан
17	17	17	2	12	2025-02-07 16:00:00.0000000	Новый
18	18	18	3	13	2025-02-08 17:00:00.0000000	Создан
19	19	19	4	14	2025-02-09 09:00:00.0000000	Отменён
20	20	20	5	15	2025-02-10 10:00:00.0000000	Завершён
21	21	21	1	16	2025-02-11 11:00:00.0000000	Новый
22	22	22	2	17	2025-02-12 12:00:00.0000000	Создан
23	23	23	3	18	2025-02-13 13:00:00.0000000	В процессе
24	24	24	4	19	2025-02-14 14:00:00.0000000	Создан
25	25	25	5	20	2025-02-15 15:00:00.0000000	Новый

Рисунок 3.43. Данные из таблицы «ProvidedServices»

3.3.3. Связи между таблицами

1) Products (Товары) → Suppliers (Поставщики)

- Тип связи: Один ко многим (1:N)
- Описание: Один поставщик может поставлять множество товаров.
- Реализация: Внешний ключ SupplierID в таблице Products ссылается на первичный ключ ID в таблице Suppliers.

2) Stock (Склад) → Products (Товары)

- Тип связи: Один к одному (1:1) (одна запись на складе для конкретного товара)

- Тип связи (с точки зрения N-арности): Один ко многим (1:N) (один товар может иметь одну запись о его остатке на складе, но таблица Stock может содержать записи для многих разных товаров).
- Описание: Каждая запись в Stock отражает остаток одного конкретного товара.
- Реализация: Внешний ключ ProductID в таблице Stock ссылается на первичный ключ ID в таблице Products.

3) Orders (Заказы) → Clients (Клиенты)

- Тип связи: Один ко многим (1:N)
- Описание: Один клиент может сделать множество заказов.
- Реализация: Внешний ключ ClientID в таблице Orders ссылается на первичный ключ ID в таблице Clients.

4) Orders (Заказы) → Employees (Сотрудники)

- Тип связи: Один ко многим (1:N)
- Описание: Один сотрудник может обслужить множество заказов.
- Реализация: Внешний ключ EmployeeID в таблице Orders ссылается на первичный ключ ID в таблице Employees.

5) OrderItems (Товары в заказе) → Orders (Заказы)

- Тип связи: Один ко многим (1:N) (обратная связь от Orders к OrderItems через N:1)
- Описание: Один заказ может содержать множество позиций (строк в OrderItems).
- Реализация: Внешний ключ OrderID в таблице OrderItems ссылается на первичный ключ ID в таблице Orders. Установлено каскадное удаление (ON DELETE CASCADE).

6) OrderItems (Товары в заказе) → Products (Товары)

- Тип связи: Один ко многим (1:N) (обратная связь от Products к OrderItems через N:1)

- Описание: Один товар может быть включён в множество позиций заказов (в разные заказы или несколько раз в одном).
- Реализация: Внешний ключ ProductID в таблице OrderItems ссылается на первичный ключ ID в таблице Products. Установлено ON DELETE NO ACTION, чтобы не позволить удалить товар, если он уже включён в позиции заказов.

7) Auth (Авторизация) → Employees (Сотрудники)

- Тип связи: Один к одному (1:1)
- Описание: У каждого сотрудника может быть только одна учётная запись в системе, и каждая учётная запись привязана к одному сотруднику.
- Реализация: Первичный ключ EmployeeID в таблице Auth одновременно является и внешним ключом, ссылающимся на первичный ключ ID в таблице Employees. Установлено каскадное удаление (ON DELETE CASCADE).

8) ProvidedServices (Оказанные услуги) → Clients (Клиенты)

- Тип связи: Один ко многим (1:N)
- Описание: Один клиент может получить множество оказанных услуг.
- Реализация: Внешний ключ ClientID в таблице ProvidedServices ссылается на первичный ключ ID в таблице Clients. Установлено каскадное удаление (ON DELETE CASCADE).

9) ProvidedServices (Оказанные услуги) → Employees (Сотрудники)

- Тип связи: Один ко многим (1:N)
- Описание: Один сотрудник может оказать множество услуг.
- Реализация: Внешний ключ EmployeeID в таблице ProvidedServices ссылается на первичный ключ ID в таблице Employees. Установлено каскадное удаление (ON DELETE CASCADE).

10) ProvidedServices (Оказанные услуги) → Services (Услуги)

- Тип связи: Один ко многим (1:N) (обратная связь от Services к ProvidedServices через N:1)

- Описание: Одна услуга может быть оказана множество раз разным клиентам разными сотрудниками.
- Реализация: Внешний ключ ServiceID в таблице ProvidedServices ссылается на первичный ключ ID в таблице Services. Установлено ON DELETE NO ACTION, чтобы не позволить удалить услугу, если она уже была оказана.

Примечания:

- Связь Products -> Stock можно трактовать как 1:1 (один товар — одна запись остатка) или как 1:N (один товар — много записей, если учитывать историю изменений остатков по времени, но в представленной структуре Stock одна запись на товар).
- Связь OrderItems -> Products (ON DELETE NO ACTION) означает, что товар нельзя будет удалить, если он есть в каком-либо заказе. Это обеспечивает целостность исторических данных заказов.
- Связь ProvidedServices -> Services (ON DELETE NO ACTION) работает аналогично, защищая историю оказанных услуг от удаления самого типа услуги.
- Связи с каскадным удалением (Orders -> Clients, Orders -> Employees, ProvidedServices -> Clients, ProvidedServices -> Employees, Auth -> Employees) означают, что при удалении клиента, сотрудника или поставщика (в случае Products -> Suppliers) будут автоматически удалены связанные с ними заказы, оказанные услуги или учётные записи (в случае Auth). Это может быть полезно, но требует осторожности при удалении записей.
- Использованы проверки (CHECK), чтобы гарантировать, что определённые поля (например, цены, количества) не могут быть отрицательными.
- В таблице OrderItems поле Subtotal — это вычисляемое persisted поле, значение которого рассчитывается автоматически при вставке/обновлении строк.

3.3.4. Индексы в БД

Индексы в базе данных представляют собой специальные структуры, предназначенные для ускорения процесса поиска и сортировки данных в таблицах. Они аналогичны указателям или оглавлению в книге, позволяя системе быстро находить нужную информацию без необходимости просмотра всех строк. Индекс создается на одном или нескольких столбцах таблицы и содержит упорядоченные значения этих столбцов, а также ссылки на соответствующие строки в основной таблице. Применение индексов значительно повышает производительность запросов, особенно при выполнении операций SELECT, JOIN, WHERE и ORDER BY. Однако их использование требует дополнительного места на диске и может замедлять операции вставки, обновления и удаления, так как индекс также необходимо обновлять при изменении данных в таблице. Поэтому важно рационально выбирать столбцы для индексации, опираясь на частоту и характер запросов к базе данных.

В разработанной базе данных для информационной системы ООО «Руссоль-центр» используются некластеризованные индексы. Эти индексы создаются отдельно от данных таблицы и ускоряют выборку данных по часто используемым полям. Ниже приведён список индексов, определённых в скрипте создания базы данных:

- IX_Suppliers_CompanyName (Таблица Suppliers): Индекс на столбце CompanyName для ускорения поиска по наименованию поставщика.
- IX_Products_SupplierID (Таблица Products): Индекс на столбце SupplierID для ускорения операций, связанных с поиском товаров по поставщику.
- IX_Products_Name (Таблица Products): Индекс на столбце Name для ускорения поиска товаров по наименованию.
- IX_Stock_ProductID (Таблица Stock): Индекс на столбце ProductID для ускорения доступа к информации об остатках конкретного товара.

- IX_Stock_LastUpdated (Таблица Stock): Индекс на столбце LastUpdated для ускорения запросов, связанных с датой последнего обновления остатков.
- IX_Clients_FullName (Таблица Clients): Индекс на столбце FullName для ускорения поиска клиентов по наименованию или ФИО.
- IX_Clients_Email (Таблица Clients): Индекс на столбце Email для ускорения поиска по адресу электронной почты.
- IX_Employees_FullName (Таблица Employees): Индекс на столбце FullName для ускорения поиска сотрудников по ФИО.
- IX_Employees_Position (Таблица Employees): Индекс на столбце Position для ускорения поиска сотрудников по должности.
- IX_Auth_Login (Таблица Auth): Индекс на столбце Login для ускорения процесса аутентификации пользователей.
- IX_Auth_AccessLevel (Таблица Auth): Индекс на столбце AccessLevel для ускорения фильтрации пользователей по уровню доступа.
- IX_Orders_ClientID (Таблица Orders): Индекс на столбце ClientID для ускорения поиска заказов по клиенту.
- IX_Orders_EmployeeID (Таблица Orders): Индекс на столбце EmployeeID для ускорения поиска заказов по сотруднику.
- IX_Orders_OrderDateTime (Таблица Orders): Индекс на столбце OrderDateTime для ускорения запросов, отсортированных по дате заказа.
- IX_Orders_Status (Таблица Orders): Индекс на столбце Status для ускорения фильтрации заказов по статусу.
- IX_OrderItems_OrderID (Таблица OrderItems): Индекс на столбце OrderID для ускорения доступа к позициям конкретного заказа.
- IX_OrderItems_ProductID (Таблица OrderItems): Индекс на столбце ProductID для ускорения поиска заказов, содержащих определённый товар.
- IX_Services_Name (Таблица Services): Индекс на столбце Name для ускорения поиска услуг по наименованию.

- IX_ProvidedServices_ClientID (Таблица ProvidedServices): Индекс на столбце ClientID для ускорения поиска оказанных услуг по клиенту.
- IX_ProvidedServices_EmployeeID (Таблица ProvidedServices): Индекс на столбце EmployeeID для ускорения поиска оказанных услуг по сотруднику.
- IX_ProvidedServices_ServiceID (Таблица ProvidedServices): Индекс на столбце ServiceID для ускорения поиска оказанных услуг по типу услуги.
- IX_ProvidedServices_ServiceDateTime (Таблица ProvidedServices): Индекс на столбце ServiceDateTime для ускорения запросов, отсортированных по дате оказания услуги.
- IX_ProvidedServices_Status (Таблица ProvidedServices): Индекс на столбце Status для ускорения фильтрации оказанных услуг по статусу.

Использование этих индексов позволяет повысить общую производительность системы за счёт ускорения типовых операций поиска и фильтрации, что особенно важно при работе с постоянно растущим объёмом данных.

3.4. Отладка и тестирование программы

Отладка и тестирование являются неотъемлемой частью процесса разработки программного обеспечения. Эти этапы играют ключевую роль в обеспечении корректности, надежности, производительности и удобства использования создаваемой информационной системы. Целью отладки и тестирования в рамках проекта для ООО «Руссоль-центр» является выявление и устранение потенциальных ошибок (дефектов, багов) в программном коде и логике работы системы до её передачи в эксплуатацию. Это позволяет гарантировать, что система будет соответствовать сформулированным требованиям (см. раздел 2.1) и корректно выполнять возложенные на неё функции, тем самым повышая эффективность бизнес-процессов компании.

3.4.1. Цель и задачи отладки и тестирования

Цель: проверить, насколько разработанное программное обеспечение соответствует заявленным функциональным и нефункциональным

требованиям, обнаружить и устранить ошибки, обеспечить стабильную и корректную работу системы в различных сценариях использования.

Задачи:

1) Выявление дефектов: найти ошибки в логике программы, взаимодействии компонентов, обработке данных и пользовательском интерфейсе.

2) Проверка функциональности: убедиться, что все функции системы работают в соответствии с требованиями (например, корректное создание заказа, изменение статуса, расчёт суммы).

3) Проверка надёжности: убедиться, что система устойчива к сбоям, корректно обрабатывает ошибки ввода и исключительные ситуации (например, попытка удалить заказ, находящийся в статусе "Отгружен").

4) Проверка производительности: убедиться, что система отвечает на запросы в разумные сроки при нормальной нагрузке (например, открытие списка заказов не занимает более 2-3 секунд).

5) Проверка безопасности: убедиться, что реализовано корректное разграничение прав доступа и защищены критичные данные.

6) Проверка юзабилити: убедиться, что интерфейс интуитивно понятен и удобен для конечных пользователей (менеджеров, администраторов).

7) Документирование результатов: зафиксировать найденные ошибки, выполненные тесты и их результаты для анализа и дальнейшего сопровождения.

3.4.2. Методы и виды тестирования

Для всесторонней проверки системы использовались следующие методы и виды тестирования:

- Ручное тестирование: в ходе этого метода тестировщик (в данном случае — разработчик) вручную выполняет подготовленные тестовые сценарии, взаимодействуя с интерфейсом приложения и проверяя его поведение. Это позволяет проверить удобство интерфейса (UI/UX тестирование), логику работы на уровне пользователя и выявить визуальные

ошибки. Этот метод особенно важен на этапе разработки отдельных модулей и при интеграционном тестировании.

- Функциональное тестирование: проверяет, соответствует ли программа заявленным функциональным требованиям. Например, при создании заказа проверяется, корректно ли заполняются поля, сохраняется ли заказ в базе данных, обновляются ли остатки на складе (если применимо).

- Интеграционное тестирование: проверяет взаимодействие между различными модулями программы, например, между пользовательским интерфейсом и слоем доступа к базе данных. Убеждается, что данные корректно передаются и обрабатываются между компонентами.

- Модульное тестирование (Unit Testing): предполагает тестирование отдельных, изолированных частей программы (методов, классов), чтобы убедиться в их корректной работе. Хотя в рамках данного проекта не были созданы автоматизированные unit-тесты, при отладке отдельных функций (например, расчёта суммы заказа) использовался принцип модульного тестирования — проверка логики конкретной части кода.

- Тестирование пользовательского интерфейса (UI Testing): проверяет, что элементы интерфейса (окна, кнопки, таблицы, поля ввода) отображаются корректно, реагируют на действия пользователя (нажатия, ввод данных) и передают данные дальше по системе.

- Регрессионное тестирование: выполняется после внесения изменений в код, чтобы убедиться, что старый функционал не сломался.

- Тестирование на граничных значениях и некорректный ввод: проверяет, как система реагирует на ввод данных, выходящих за пределы ожидаемых значений (например, отрицательное количество товара в заказе) или некорректных данных (например, буквы в поле "Количество"). Система должна корректно обрабатывать такие ситуации, не падая, а выдавая понятное сообщение об ошибке.

3.4.3. Инструменты отладки и тестирования

В процессе отладки и тестирования использовались инструменты, предоставляемые средой разработки и операционной системой:

Встроенный отладчик Microsoft Visual Studio Community: Один из самых мощных и удобных инструментов. Он позволяет:

- Устанавливать точки останова (breakpoints) в коде, чтобы приложение приостанавливало выполнение и позволяло изучить текущее состояние переменных, полей, объектов.
- Пошагово выполнять код (Step Into, Step Over, Step Out) для детального анализа логики.
- Наблюдать за значениями переменных в реальном времени.
- Проверять стек вызовов функций.
- Исследовать исключения и их источники. Это позволяло локализовать и исправить логические ошибки в алгоритмах работы программы.

SQL Server Management Studio (SSMS): использовался для проверки корректности работы SQL-запросов, вставки, обновления и удаления данных в базе данных вручную, а также для анализа состояния таблиц и отладки сложных запросов, используемых в приложении.

Самотестирование и проверка вручную: как описано выше, значительная часть тестирования проводилась вручную, с использованием подготовленных сценариев и проверкой результатов визуально и через анализ данных в базе.

3.4.4. Проведение тестирования

Тестирование проводилось на этапе разработки и после завершения основного цикла программирования.

1) Подготовка тестов: были разработаны тестовые сценарии, охватывающие основные функции системы:

Сценарий 1: Авторизация пользователя. Проверка входа с корректными и некорректными логином/паролем.

Сценарий 2: Проверка разграничения прав. Вход под разными ролями (Администратор, Менеджер по заказам, Менеджер по услугам) и проверка доступности соответствующих функций в главном меню.

Сценарий 3: Работа с заказами. Создание нового заказа, проверка расчёта суммы, изменение статуса, удаление заказа.

Сценарий 4: Работа с оказанными услугами. Создание записи об услуге, изменение статуса.

Сценарий 5: Работа со справочниками. Добавление, редактирование и удаление записей (например, товара, клиента).

Сценарий 6: Поиск и фильтрация данных в списках (заказы, клиенты, товары).

Сценарий 7: Ввод некорректных данных (пустые поля, отрицательные числа там, где они недопустимы). Проверка обработки ошибок.

Сценарий 8: Проверка работы с базой данных (проверка корректности записи, обновления и удаления данных, проверка связей и ограничений).

2) Выполнение тестов: Тестовые сценарии выполнялись вручную. Для каждого сценария фиксировался:

- Входные данные (например, введённые логин/пароль, данные заказа).
- Шаги выполнения (последовательность действий пользователя).
- Ожидаемый результат (например, успешный вход, создание заказа, отображение сообщения об ошибке).
- Фактический результат (то, что реально произошло в системе).
- Статус теста (Пройден / Не пройден / Требует уточнения).

3) Анализ результатов: после выполнения тестов анализировались результаты. Все обнаруженные ошибки (например, падение приложения, некорректный расчёт, недоступность функции) документировались (в уме или в заметках), а затем устранялись в процессе отладки.

4) Повторное тестирование: после устранения ошибок соответствующие тесты запускались заново, чтобы убедиться, что проблема действительно решена и не появились новые ("регресс").

3.4.5. Результаты тестирования

В результате проведённого тестирования были выявлены и устранены следующие типы ошибок:

- Логические ошибки в алгоритмах: например, ошибка в расчёте итоговой суммы заказа или неправильная сортировка данных.
- Ошибки обработки исключений: Программа стала более устойчивой к некорректному вводу данных (например, теперь она корректно обрабатывает попытку ввода букв в числовые поля).
- Ошибки взаимодействия с базой данных: Исправлены проблемы с некорректным обновлением данных или нарушением целостности связей при определённых действиях.
- Ошибки в интерфейсе: Исправлены мелкие визуальные баги, улучшена отзывчивость элементов управления.

Тестирование подтвердило, что основные функции системы работают корректно, требования к функциональности и удобству использования в целом выполнены. Система стабильно работает при нормальной эксплуатации и корректно реагирует на большинство ошибок ввода. Таким образом, этап отладки и тестирования позволил создать более надёжный и качественный продукт, готовый к передаче пользователю.

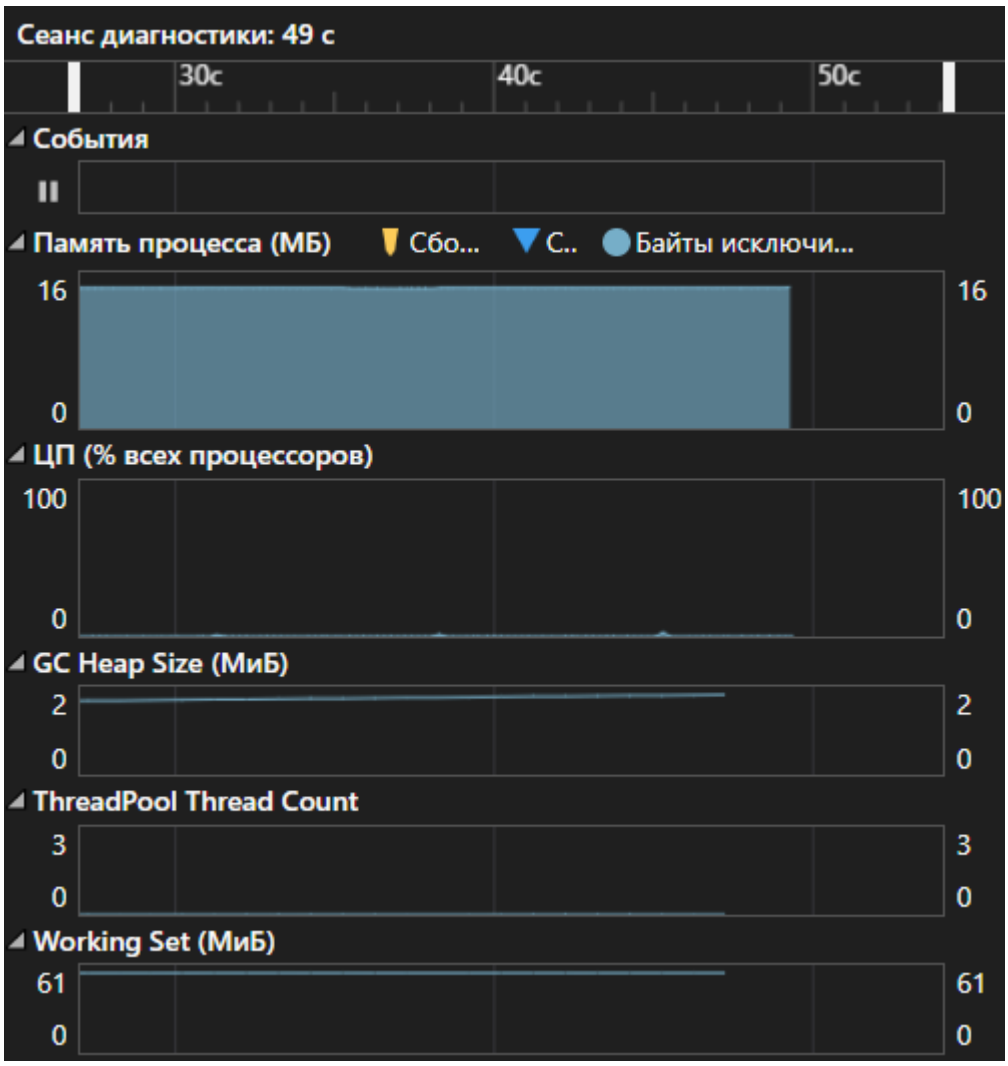

Рисунок 3.44. Производительность программы, используемые ресурсы

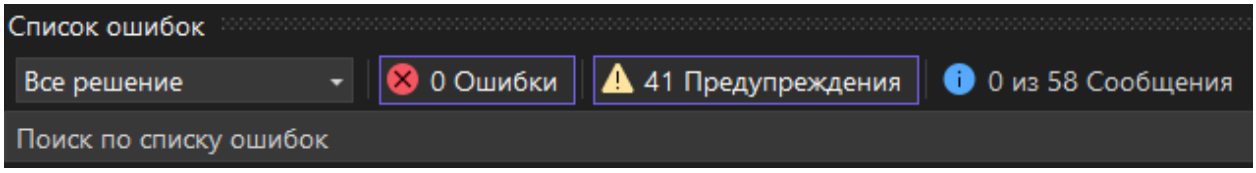

Рисунок 3.45. Список ошибок

3.5 Руководство пользователя

Настоящее руководство описывает порядок работы с информационной системой для ООО «Руссоль-центр». Оно предназначено для администраторов, менеджеров по заказам и менеджеров по услугам, которые используют систему для выполнения своих должностных обязанностей.

Цель системы: Автоматизация ключевых бизнес-процессов, включая управление заказами, оказанием услуг, складскими запасами, клиентами, поставщиками, товарами и сотрудниками.

Назначение и условия применения: Система предназначена для внутреннего использования сотрудниками ООО «Руссоль-центр». Для работы с системой необходим компьютер под управлением операционной системы Windows с установленным приложением. Пользователь должен обладать базовыми навыками работы с операционной системой Windows и иметь учётную запись в системе.

Установка и запуск: Приложение поставляется в виде исполняемого файла (*.exe). Для установки достаточно скопировать файл на локальный диск компьютера и запустить. При первом запуске или при необходимости обновления структуры базы данных может появиться запрос на подключение к серверу базы данных SQL Server. После запуска откроется окно авторизации.

Порядок работы:

1) Авторизация:

- При запуске приложения появится окно авторизации (см. рис. 2.6).
- Введите свой Логин и Пароль.
- Нажмите кнопку Войти.
- Если данные введены верно, откроется Главное окно системы. В противном случае, появится сообщение об ошибке, и потребуется повторный ввод.

2) Главное окно:

- После входа открывается Главное окно (см. рис. 2.8).
- В верхней части отображается приветствие с именем вошедшего пользователя.
- Набор кнопок в центре окна зависит от роли пользователя (Администратор, Менеджер по заказам, Менеджер по услугам). Каждая кнопка ведёт к соответствующему разделу системы.

3) Работа с функциями системы:

Работа с заказами (для Менеджера по заказам и Администратора):

- Нажмите кнопку Заказы.
- Откроется окно списка заказов (см. рис. 2.10).

- Здесь можно просматривать список заказов, фильтровать и искать.
- Для создания нового заказа нажмите Добавить.
- Для редактирования существующего заказа выберите его в списке и нажмите Изменить.
- В открывшемся окне редактирования заказа (см. рис. 2.11) выберите клиента, укажите товары и количество, система рассчитает сумму. Нажмите. Добавить для сохранения.

Работа с оказанными услугами (для Менеджера по услугам и Администратора):

- Нажмите кнопку Оказанные услуги.
- Откроется окно списка оказанных услуг.
- Для добавления новой услуги нажмите Добавить.
- Для редактирования выберите запись и нажмите Изменить.
- В открывшемся окне выберите клиента, сотрудника, услугу, укажите дату и статус. Нажмите. Добавить для сохранения.

Работа со справочниками (для Администратора и других ролей, где разрешено):

- Кнопки Клиенты, Товары, Сотрудники, Поставщики открывают соответствующие окна со списками.
- В этих окнах можно просматривать, искать, добавлять (Добавить), изменять (Изменить) и удалять (Удалить) записи.

Работа со складом (для Администратора):

- Нажмите кнопку Склад.
- Откроется окно списка остатков.
- Здесь можно просматривать текущие остатки, добавлять новые позиции или изменять количество (Добавить/Изменить).

Управление пользователями (для Администратора):

- Нажмите кнопку Авторизация.
- Откроется окно списка учётных записей.

- Здесь можно добавлять новых пользователей, редактировать их данные и права доступа (Добавить/Изменить).

Сообщения и ошибки: Система выводит понятные сообщения об ошибках (например, при вводе неверных данных или отсутствии связи с базой данных). Следуйте инструкциям в этих сообщениях.

Выход из системы: для выхода из системы закройте главное окно программы.

3.6 Руководство программиста

Руководство программиста (Developer Guide) — это документ, предназначенный для разработчиков, сопровождающих и потенциально расширяющих функциональность разработанного программного обеспечения. Он содержит техническую информацию, необходимую для понимания архитектуры, структуры кода, взаимодействия компонентов и процесса разработки. Цель руководства — облегчить дальнейшую работу с кодовой базой, упростить внесение изменений, исправление ошибок и добавление новых функций. В рамках дипломного проекта руководство программиста описывает разработанную информационную систему для ООО «Руссоль-центр», её архитектуру, используемые технологии, структуру исходного кода и особенности взаимодействия с базой данных.

Назначение и условия применения программы

Назначение: Руководство программиста предназначено для:

Ознакомления: Предоставления разработчику (включая первоначального автора) полной информации о структуре, логике и реализации программного обеспечения.

Сопровождения: Обеспечения возможности эффективного поиска и устранения ошибок (отладки), понимания влияния изменений и внесения корректировок в существующий код.

Расширения: Обеспечения понимания архитектуры и принципов проектирования, необходимых для добавления новых функций и модулей в будущем.

Передачи знаний: Обеспечения возможности передачи проекта другому разработчику без потери контекста и необходимости полного повторного анализа кода.

Область применения: Документ применяется при разработке, тестировании, сопровождении и модификации информационной системы.

Технические требования к среде разработки:

Операционная система: Microsoft Windows 10/11 (для разработки и отладки).

Среда разработки: Microsoft Visual Studio Community (или Enterprise) последних версий, поддерживающая .NET Framework 4.8 или .NET 6/7/8 (в зависимости от выбранной целевой платформы проекта).

Язык программирования: C#.

СУБД: Microsoft SQL Server (Express Edition или выше).

Средства доступа к СУБД в IDE: Встроенные инструменты Visual Studio (Server Explorer, SQL Server Object Explorer) или SQL Server Management Studio (SSMS) для ручного управления БД.

Система контроля версий (желательно): Git (интегрирован в Visual Studio).

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была разработана информационная система для автоматизации деятельности ООО «Руссоль-центр». Проект охватывал ключевые этапы жизненного цикла программного обеспечения: анализ предметной области, постановку задачи, проектирование архитектуры и базы данных, разработку интерфейса, программную реализацию и тестирование.

Цель работы, состоявшаяся в разработке программного обеспечения для автоматизации учёта заказов, оказанных услуг, складских запасов, клиентов, поставщиков, товаров и сотрудников, была достигнута. Для этого были решены следующие задачи:

Анализ предметной области: была проведена детальная диагностика бизнес-процессов компании, выявлены слабые места в существующих ручных или частично автоматизированных операциях (например, трудности с учётом остатков, поиском информации, контролем статуса заказов), что подтвердило актуальность темы.

Сравнительный анализ систем-аналогов: были рассмотрены существующие решения (включая 1С, готовые CRM и WMS). Анализ показал, что универсальные или сложные корпоративные системы либо избыточны, либо не учитывают специфику деятельности ООО «Руссоль-центр», что обосновало необходимость создания индивидуального решения.

Формирование требований: на основе анализа были сформулированы и зафиксированы как функциональные (управление заказами, услугами, справочниками, складом, отчётность), так и нефункциональные требования (производительность, безопасность, удобство использования), определены акторы и их роли (Администратор, Менеджер по заказам, Менеджер по услугам).

Проектирование системы: были созданы модели данных (ER-диаграмма), модели поведения (DFD, IDEF0, Use Case), макеты интерфейса,

определена архитектура приложения (клиент-серверная, слоистая) и выбраны средства разработки (C#, .NET, Windows Forms, SQL Server).

Разработка программного обеспечения: Реализована логика функционирования системы, включая интерфейс пользователя, слой доступа к данным, бизнес-логику. Программный продукт обеспечивает выполнение всех запланированных функций в соответствии с требованиями.

Тестирование: Проведено тестирование ключевых функций, подтвердившее работоспособность системы и соответствие основным требованиям.

Практическое значение разработанной системы заключается в том, что она позволяет:

- Повысить эффективность бизнес-процессов за счёт автоматизации рутинных операций.
- Сократить время на поиск информации, формирование документов и отчётов.
- Повысить точность учёта и снизить вероятность ошибок.
- Обеспечить централизованное хранение и контроль над данными.
- Обеспечить разграничение доступа к информации в зависимости от роли пользователя.

Таким образом, в результате выполнения дипломного проекта создана функциональная, надёжная и удобная информационная система, адаптированная под специфику деятельности ООО «Руссоль-центр». Работа продемонстрировала способность автора применить теоретические знания и практические навыки в области анализа, проектирования и программирования для решения реальной прикладной задачи, что подтверждает её научно-практическую значимость и достижение поставленной цели.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Назаров, С.В. Программирование на C# и .NET: от основ до практики / Сергей Викторович Назаров. — М.: ДМК Пресс, 2023. — 672 с. — ISBN 978-5-97061-228-8.

2. Петренко, Д.Н. Разработка Windows-приложений: от идеи до релиза / Дмитрий Николаевич Петренко. — М.: Эксмо, 2023. — 528 с. — ISBN 978-5-04-118845-6.

3. Мамедли Р.Э. Системы управления базами данных: учебник для СПО / Р.Э. Мамедли. — СПб.: Лань, 2024. — 228 с. — ISBN 978-5-8114-7961-9.

Электронные ресурсы:

1. Гаврилов М.В. Информатика и информационные технологии: учебник для вузов / М.В. Гаврилов, В.А. Климов. — 5-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2024. — 355 с. — ISBN 978-5-534-13932-1. — URL: http://biblio-online.ru/book/informatika-i-informacionnye-tehnologii-449286 (дата обращения: 15.11.2024). — Текст: электронный.

2. Плещев В.В. Основы программирования на языках C++ и C# с примерами и упражнениями: учеб. пособие / В. В. Плещев, Е. И. Шишков. — 2021. — 286 с. — URL: https://bookview.ru/books/2021_osnovy_programmirovaniya_na_yazykah_c_i_c_s_primerami_i_uprazhneniyami_ucheb_posobie_pl/ (дата обращения: 20.10.2024). — Текст: электронный.

3. Мартыненко Т.В. Основы визуального программирования в среде VISUAL STUDIO на базе C#: учебное пособие / Т.В. Мартыненко, В.В. Турупалов, Н.К. Андриевская. — Вологда: Инфра-Инженерия, 2023. — 232 с. — ISBN 978-5-99858-214-1. — Текст: электронный.

4. Рочев К.В. Информационные технологии. Анализ и проектирование информационных систем: учебное пособие для вузов / К.В. Рочев. — 3-е изд., стер. — Санкт-Петербург: Лань, 2022. — 128 с. — ISBN 978-5-8114-6302-1. — URL: https://bookview.ru/books/2022_informacionnye_tehnologii_analiz_i_proektirovan

ie_informacionnyh_sistem_ucheb_posobie_dlya_vuzov_rochev_k_v_/ (дата обращения: 05.11.2024). — Текст: электронный.

5. Концепция пользовательского интерфейса. – URL: https://v8.1c.ru/platforma/kontseptsiya-polzovatelskogo-interfeysa/ (дата обращения: 02.11.2024). — Текст: электронный.

6. Microsoft SQL Server Documentation. – URL: https://docs.microsoft.com/en-us/sql/sql-server/ (дата обращения: 10.11.2024). — Текст: электронный.

7. Microsoft .NET Documentation. – URL: https://docs.microsoft.com/en-us/dotnet/ (дата обращения: 18.11.2024). — Текст: электронный.

ПРИЛОЖЕНИЕ

Класс Program.cs

```
namespace Program.scr
{
    internal static class Program
    {
        /// <summary>
        ///  The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            // To customize application configuration such as set high DPI settings
or default font,
            // see https://aka.ms/applicationconfiguration.
            ApplicationConfiguration.Initialize();
            Application.Run(new AuthForm());
        }
    }
}
```

Класс LabeledNumericItem.cs

```
namespace Program.scr.windows.customListBox
{
    public class LabeledNumericItem : UserControl
    {
        public int Id { get; private set; }
        public string Key { get; private set; }
        public decimal Value => numericUpDown.Value;

        public NumericUpDown numericUpDown;

        public LabeledNumericItem(int id, string key, decimal value, decimal maximum,
Action<int, string, decimal> onValueChanged)
        {
            Id = id;
            Key = key;
```

```
            this.Height = 30;
            this.Width = 1000;
            this.AutoSize = false;

            var label = new Label
            {
                Text = key,
                Width = 390,
                Height = 25,
                Dock = DockStyle.Left,
                AutoEllipsis = true,
                //AutoSize = true,
                TextAlign = ContentAlignment.MiddleLeft,
                Margin = new Padding(3),
                Anchor = AnchorStyles.Left | AnchorStyles.Top
            };

            numericUpDown = new NumericUpDown
            {
                DecimalPlaces = 4,
                Minimum = 0,
                Maximum = Math.Max(0, maximum), // Ограничиваем минимумом, чтобы не
было <= 0
                Increment = 1.0000m,
                Value = Math.Min(Math.Max(0, value), maximum), // Значение в пределах
[min, max]
                Width = 120,
                Height = 25,
                Location = new Point(400, 0),
                Dock = DockStyle.Right,
                Margin = new Padding(3),

                Anchor = AnchorStyles.Right | AnchorStyles.Top
            };
            numericUpDown.ValueChanged += (object? sender, EventArgs e) =>
            {
                numericUpDown.BackColor = numericUpDown.Maximum == 0 ?
Color.IndianRed : Color.White;
```

```
                numericUpDown.BackColor = numericUpDown.Value == 0 ? Color.White :
Color.LightGreen;
                onValueChanged?.Invoke(Id, Key, numericUpDown.Value);
            };

            this.Controls.Add(numericUpDown);
            this.Controls.Add(label);

            numericUpDown.BackColor = numericUpDown.Value == 0 ? Color.White :
Color.LightGreen;
        }



        public void SetValue(decimal value)
        {
            if (value < 0) value = 0;
            if (value > numericUpDown.Maximum) value = numericUpDown.Maximum;
            numericUpDown.Value = Math.Round(value, 4);
        }

        public void SetMaximum(decimal maximum)
        {
            if (maximum <= 0) maximum = 1.0000m;
            numericUpDown.Maximum = Math.Round(maximum, 4);
            if (numericUpDown.Value > numericUpDown.Maximum)
                numericUpDown.Value = numericUpDown.Maximum;
        }
    }
}
```

Класс CustomNumericListBox.cs

```
namespace Program.scr.windows.customListBox
{
    public class CustomNumericListBox : Panel
    {
        private FlowLayoutPanel flowPanel;
        private List<LabeledNumericItem> items;
```

```
        public event EventHandler<ValueChangedEventArgs> ValueChanged;

        public IReadOnlyList<LabeledNumericItem> Items => items.AsReadOnly();

        public CustomNumericListBox()
        {
            items = new List<LabeledNumericItem>();

            flowPanel = new FlowLayoutPanel
            {
                Dock = DockStyle.Fill,
                AutoScroll = true,
                FlowDirection = FlowDirection.TopDown,
                WrapContents = false
            };

            this.Controls.Add(flowPanel);
            this.BorderStyle = BorderStyle.FixedSingle;
        }

        // Добавление с ID
        public void Add(int id, string key, decimal value, decimal maxValue =
1000.0000m)
        {
            // Лямбда вызывает наше событие
            void OnValueChanged(int itemId, string itemKey, decimal newValue)
            {
                ValueChanged?.Invoke(this, new ValueChangedEventArgs(itemId, itemKey,
newValue));
            }

            var item = new LabeledNumericItem(id, key, value, maxValue,
OnValueChanged);
            items.Add(item);
            flowPanel.Controls.Add(item);
        }

        // Получение значения по ID
```

```
    public decimal? GetValueById(int id)
    {
        var item = items.FirstOrDefault(i => i.Id == id);
        return item?.Value;
    }

    // Опционально: получение элемента по ID
    public LabeledNumericItem GetItemById(int id)
    {
        return items.FirstOrDefault(i => i.Id == id);
    }

    public void Clear()
    {
        flowPanel.Controls.Clear();
        items.Clear();
    }

    public void RemoveAt(int index)
    {
        if (index >= 0 && index < items.Count)
        {
            flowPanel.Controls.Remove(items[index]);
            items.RemoveAt(index);
        }
    }
}

public class ValueChangedEventArgs : EventArgs
{
    public int Id { get; }
    public string Key { get; }
    public decimal Value { get; }

    public ValueChangedEventArgs(int id, string key, decimal value)
    {
        Id = id;
```

```
            Key = key;
            Value = value;
        }
    }
}
```

Класс Orders_AddEditForm.cs

```
using Program.scr.core.dbt;
using Program.scr.windows.customListBox;
namespace Program.scr.windows
{
    public partial class Orders_AddEditForm : Form
    {
        DBT_Orders Object;

        Button button_apply;
        TableLayoutPanel tableLayout;

        ComboBox comboBox_ClientID;
        ComboBox comboBox_EmployeeID;
        DateTimePicker dateTimePicker_OrderDateTime;
        TextBox textBox_TotalAmount;
        TextBox textBox_Status;
        CustomNumericListBox customListBox;

        private List<DBT_Clients> Clients = new List<DBT_Clients>();
        private List<DBT_Employees> Employees = new List<DBT_Employees>();

        public Orders_AddEditForm()
        {
            InitializeComponent();
            Init();

            foreach (var i in DBT_Products.GetAll()) customListBox.Add(i.ID, i.Name,
0, DBT_Stock.GetByProductID(i.ID).QuantityOnStock);
        }
        public Orders_AddEditForm(DBT_Orders obj)
```

```
        {
            InitializeComponent();
            Object = obj;
            Init();

            comboBox_ClientID.SelectedIndex = indexOf_Clients(obj.ClientID);
            comboBox_EmployeeID.SelectedIndex = indexOf_Employees(obj.EmployeeID);
            dateTimePicker_OrderDateTime.Value = (DateTime)((obj.OrderDateTime ==
null) ? DateTime.Now : obj.OrderDateTime);
            decimal total = 0;
            foreach (var i in DBT_OrderItems.GetByOrderID(obj.ID))
            {
                var product = DBT_Products.GetById(i.ProductID);
                total += (decimal)i.Subtotal;
            }
            textBox_TotalAmount.Text = total.ToString();
            textBox_Status.Text = obj.Status.ToString();

            foreach (var i in DBT_Products.GetAll())
            {
                var result = DBT_OrderItems.GetByOrderIDAndProductID(obj.ID, i.ID);
                decimal value = 0;
                if (result != null) if (result.Count > 0) if (result[0] != null)
value = result[0].Quantity;
                customListBox.Add(i.ID, i.Name, value,
DBT_Stock.GetByProductID(i.ID).QuantityOnStock + value);
            }

            //foreach (var i in customListBox.Items) i.numericUpDown.Enabled = false;
            //comboBox_ClientID.Enabled = false;
            //comboBox_EmployeeID.Enabled = false;
        }

        private void Init()
        {
            this.Size = new Size(1200, 650);
            this.Text = "Заказы - " + (Object == null ? "Добавить" :
"Изменить").ToString();
```

```
this.MinimumSize = new Size(400, 400);
this.StartPosition = FormStartPosition.CenterScreen;
button_apply = new Button()
{
    Height = 30,
    Dock = DockStyle.Bottom
};
button_apply.Click += Button_apply_Click;
button_apply.Text = Object == null ? "Добавить" : "Изменить";
this.Controls.Add(button_apply);

tableLayout = new TableLayoutPanel()
{
    Dock = DockStyle.Top,
    Height = 190,
    ColumnCount = 2,
    RowCount = 6
};

Label label_ClientID = new Label();
SetLabel(ref label_ClientID, "Клиент");
tableLayout.Controls.Add(label_ClientID, 0, 0);
comboBox_ClientID = new ComboBox();
comboBox_ClientID.Dock = DockStyle.Fill;
comboBox_ClientID.MaxLength = 254;
tableLayout.Controls.Add(comboBox_ClientID, 1, 0);

Label label_EmployeeID = new Label();
SetLabel(ref label_EmployeeID, "Сотрудник");
tableLayout.Controls.Add(label_EmployeeID, 0, 1);
comboBox_EmployeeID = new ComboBox();
comboBox_EmployeeID.Dock = DockStyle.Fill;
comboBox_EmployeeID.MaxLength = 254;
tableLayout.Controls.Add(comboBox_EmployeeID, 1, 1);

Label label_OrderDateTime = new Label();
SetLabel(ref label_OrderDateTime, "Дата заказа");
```

```
tableLayout.Controls.Add(label_OrderDateTime, 0, 2);
dateTimePicker_OrderDateTime = new DateTimePicker();
dateTimePicker_OrderDateTime.Dock = DockStyle.Fill;
tableLayout.Controls.Add(dateTimePicker_OrderDateTime, 1, 2);
dateTimePicker_OrderDateTime.Format = DateTimePickerFormat.Custom;
dateTimePicker_OrderDateTime.CustomFormat = "yyyy.MM.dd HH:mm:ss";

Label label_TotalAmount = new Label();
SetLabel(ref label_TotalAmount, "Сумма заказа");
tableLayout.Controls.Add(label_TotalAmount, 0, 3);
textBox_TotalAmount = new TextBox();
textBox_TotalAmount.ReadOnly = true;
textBox_TotalAmount.Dock = DockStyle.Fill;
textBox_TotalAmount.MaxLength = 254;
tableLayout.Controls.Add(textBox_TotalAmount, 1, 3);

Label label_Status = new Label();
SetLabel(ref label_Status, "Статус");
tableLayout.Controls.Add(label_Status, 0, 4);
textBox_Status = new TextBox();
textBox_Status.Dock = DockStyle.Fill;
textBox_Status.MaxLength = 254;
tableLayout.Controls.Add(textBox_Status, 1, 4);

Label label_Products = new Label();
SetLabel(ref label_Products, "Товары");
tableLayout.Controls.Add(label_Products, 0, 5);

customListBox = new CustomNumericListBox();
customListBox.Dock = DockStyle.Fill;
customListBox.Width = 400;
customListBox.ValueChanged += (sender, e) =>
{
    decimal total = 0;
    foreach (var i in customListBox.Items)
    {
        total += i.Value * DBT_Products.GetById(i.Id).PricePerUnit;
```

```
        }
        total = Math.Round(total, 2);
        textBox_TotalAmount.Text = total.ToString();
    };
    this.Controls.Add(customListBox);



    this.Controls.Add(tableLayout);
    this.Controls.Add(button_apply);

    LoadComboBox_Clients();
    LoadComboBox_Employees();
}

private void LoadComboBox_Clients()
{
    Clients = DBT_Clients.GetAll();

    comboBox_ClientID.Items.Clear();
    foreach (var i in Clients)
        comboBox_ClientID.Items.Add(i.FullName);
}
private int indexOf_Clients(int id)
{
    for (int i = 0; i < Clients.Count; i++)
    {
        if (Clients[i].ID == id) return i;
    }
    return -1;
}
private void LoadComboBox_Employees()
{
    Employees = DBT_Employees.GetAll();

    comboBox_EmployeeID.Items.Clear();
    foreach (var i in Employees)
        comboBox_EmployeeID.Items.Add(i.FullName);
```

```
        }
        private int indexOf_Employees(int id)
        {
            for (int i = 0; i < Employees.Count; i++)
            {
                if (Employees[i].ID == id) return i;
            }
            return -1;
        }

        private void SetLabel(ref Label label, string text = "")
        {
            label.Font = new Font(Font.FontFamily, 12);
            label.TextAlign = ContentAlignment.TopLeft;
            label.Dock = DockStyle.Fill;
            label.AutoSize = false;
            label.Width = 200;
            label.Text = text;
        }

        private void Button_apply_Click(object? sender, EventArgs e)
        {
            if (comboBox_ClientID.SelectedIndex == -1) { MessageBox.Show("Поле
'Клиент' имеет некорректное значение!"); return; }
            if (comboBox_EmployeeID.SelectedIndex == -1) { MessageBox.Show("Поле
'Сотрудник' имеет некорректное значение!"); return; }
            if (!decimal.TryParse(textBox_TotalAmount.Text, out decimal
tp_TotalAmount)) { MessageBox.Show("Поле 'Сумма заказа' имеет некорректное
значение!"); return; }
            if (string.IsNullOrWhiteSpace(textBox_Status.Text))
{ MessageBox.Show("Поле 'Статус' имеет некорректное значение!"); return; }

            int res = 0;

            if (Object == null)
            {
                res = DBT_Orders.Create(
                    new DBT_Orders()
                    {
```

```
                        ClientID = Clients[comboBox_ClientID.SelectedIndex].ID,
                        EmployeeID = Employees[comboBox_EmployeeID.SelectedIndex].ID,
                        OrderDateTime = dateTimePicker_OrderDateTime.Value,
                        TotalAmount = decimal.Parse(textBox_TotalAmount.Text),
                        Status = textBox_Status.Text
                    }
                );
                if (res == -1)
                {
                    MessageBox.Show("Ошибка! Один из ID не ссылается на запись в
БД!");
                    return;
                }

                foreach (var i in customListBox.Items)
                {
                    if (i.Value == 0) continue;

                    // списать со склада
                    DBT_Stock.DebitFromWarehouse(i.Id, i.Value);

                    // добавить записи в orderItems
                    DBT_OrderItems.Create(
                        new DBT_OrderItems()
                        {
                            OrderID = res,
                            ProductID = i.Id,
                            Quantity = i.Value,
                            PriceAtOrderTime =
DBT_Products.GetById(i.Id).PricePerUnit
                        }
                        );
                }
            }
            else
            {
                res = DBT_Orders.Edit(
                    new DBT_Orders()
```

```
                    {
                        ID = Object.ID,
                        ClientID = Clients[comboBox_ClientID.SelectedIndex].ID,
                        EmployeeID = Employees[comboBox_EmployeeID.SelectedIndex].ID,
                        OrderDateTime = dateTimePicker_OrderDateTime.Value,
                        TotalAmount = decimal.Parse(textBox_TotalAmount.Text),
                        Status = textBox_Status.Text
                    }
                );

                // вернуть всё на склад по данным из БД
                foreach (var i in DBT_OrderItems.GetByOrderID(Object.ID))
                {
                    DBT_Stock.DebitFromWarehouse(i.ProductID, -i.Quantity);
                }

                // удалить связи
                DBT_OrderItems.RemoveByOrderId(Object.ID);

                foreach (var i in customListBox.Items)
                {
                    if (i.Value == 0) continue;

                    // списать со склада
                    DBT_Stock.DebitFromWarehouse(i.Id, i.Value);

                    // добавить записи в orderItems
                    DBT_OrderItems.Create(
                        new DBT_OrderItems()
                        {
                            OrderID = Object.ID,
                            ProductID = i.Id,
                            Quantity = i.Value,
                            PriceAtOrderTime =
DBT_Products.GetById(i.Id).PricePerUnit
                        }
                        );
```

```
                }
            }
            if (res == -1) MessageBox.Show("Ошибка! Один из ID не ссылается на запись
в БД!");
            else this.Close();
        }

        public void DebitFromWarehouse()
        {

        }
    }
}
```