

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 системы отдела продаж салонов сотовой связи»**

Выполнила:
студентка группы ИСП-О-22
Котельников Владислав Вусалович

(подпись)

РАБОТА ДОПУЩЕНА К ЗАЩИТЕ
Заместитель директора по УР
М.В. Казакова

(подпись)

МП

Руководитель:
преподаватель колледжа,
Титов Андрей Дмитриевич

(подпись)

пос. Электроизолятор
2025 год

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1. АНАЛИТИЧЕСКАЯ ЧАСТЬ	9
1.1. Описание и анализ предметной области	9
1.2. Принципы разработки ИС.....	14
1.3. Характеристика проблемы разработки	16
1.4. Обзор и сравнительный анализ готовых решений	20
2. ПРОЕКТНАЯ ЧАСТЬ	25
2.1. Определение требований к информационной системе	25
2.2. Обзор и выбор средств проектирования и разработки ИС	29
2.3. Методы моделирования системы	32
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)	47
2.4. Макеты интерфейса приложения	52
2.5. Описание логики функционирования ПО	74
2.5.1. Вход в систему и разграничение прав	75
2.5.2. Работа с клиентами	76
2.5.3. Работа с продажами и контрактами	77
2.5.4. Работа со справочниками и аналитикой	77
3. СПЕЦИАЛЬНАЯ ЧАСТЬ.....	79
3.1. Формирование требований к ПО.....	79
3.2. Разработка интерфейса ПО	82
3.3. Проектирование структуры базы данных.....	89
3.3.1. Физическая ER диаграмма	93
3.3.2. Описание таблиц	93
3.3.3. Связи между таблицами	106
3.3.4. Индексы в БД.....	108

3.4. Отладка и тестирование программы	109
3.5 Руководство пользователя.....	112
ЗАКЛЮЧЕНИЕ	117
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	119
ПРИЛОЖЕНИЕ	121

ВВЕДЕНИЕ

В современном мире, где сфера услуг и розничная торговля находятся под постоянным давлением быстро меняющихся рыночных условий, высоких требований со стороны потребителей и жесткой конкуренции, эффективное функционирование бизнеса становится невозможным без внедрения современных информационных технологий. Особенно это актуально для салонов сотовой связи, где требуется оперативно управлять ассортиментом товаров, тарифными планами, данными клиентов, процессами продаж и заключения контрактов, а также обеспечивать высокий уровень обслуживания. Ручной учет и ведение бумажной документации не только требуют значительных временных и трудовых затрат, но и подвержены человеческим ошибкам, что снижает общую эффективность работы отдела продаж, усложняет аналитику и принятие управленческих решений.

Актуальность темы данной выпускной квалификационной работы обусловлена необходимостью автоматизации и оптимизации бизнес-процессов отдела продаж салонов сотовой связи. Создание специализированной информационной системы позволяет централизованно управлять данными, повышать точность и скорость обработки информации, оптимизировать взаимодействие между различными подразделениями (например, складом, бухгалтерией, отделом кадров), а также улучшать качество обслуживания клиентов за счет более быстрого и точного поиска и предоставления информации. В условиях высокой конкуренции, внедрение качественной информационной системы становится важным конкурентным преимуществом, позволяющим оперативно реагировать на изменения рынка, управлять запасами, отслеживать эффективность сотрудников и анализировать продажи. В условиях цифровизации экономики, такие системы становятся не просто инструментом, а необходимым элементом инфраструктуры современного бизнеса.

Целью выпускной квалификационной работы является разработка информационной системы отдела продаж салонов сотовой связи,

обеспечивающей эффективное управление данными клиентов, сотрудников, товаров, тарифных планов, продаж и контрактов, а также автоматизацию ключевых бизнес-процессов, включая начальный анализ, консультации, оформление сделок и формирование отчетности.

Для достижения поставленной цели в работе были решены следующие задачи:

1) Провести анализ предметной области и существующих решений (систем-аналогов). Это включало изучение типичных бизнес-процессов в салонах сотовой связи, выявление проблем и неэффективностей в текущей системе учета и управления, а также сравнение функциональности и применимости готовых решений (например, CRM-систем, специализированных розничных решений).

2) Сформировать требования к разрабатываемой информационной системе. На основе анализа были определены функциональные (что система должна уметь делать) и нефункциональные (какими качествами должна обладать система, например, надежность, производительность, удобство использования) требования.

3) Обосновать выбор средств реализации. Были рассмотрены и обоснованы выбор языка программирования, среды разработки, системы управления базами данных (СУБД) и других технологий, используемых при создании системы, с учетом поставленных задач, требований и предпочтений разработчика.

4) Спроектировать архитектуру приложения и базу данных. На этом этапе была разработана структура программного обеспечения, определены основные компоненты и их взаимодействие, а также спроектирована логическая и физическая модель базы данных для хранения информации.

5) Разработать функциональные диаграммы (DFD, IDEF0), диаграммы вариантов использования (Use Case) и другие необходимые модели. Графические модели позволяют наглядно представить процессы, потоки

данных, взаимодействие пользователей с системой и взаимосвязи между различными элементами системы.

6) Создать макеты пользовательского интерфейса. Проектирование удобного и интуитивно понятного интерфейса является важным этапом, так как от этого зависит эффективность работы пользователей и их удовлетворенность системой.

7) Осуществить программную реализацию отдельных модулей системы. Была выполнена практическая реализация ключевых функций системы, например, модулей работы с клиентами, товарами, оформления продаж и контрактов.

8) Подготовить руководство пользователя и программиста. Документация необходима для обеспечения корректного использования системы конечными пользователями и понимания её архитектуры и функционирования разработчиками.

Объектом исследования является деятельность отдела продаж салонов сотовой связи, включающая процессы продажи товаров и услуг, взаимодействия с клиентами, складом, бухгалтерией и другими внутренними подразделениями, а также формирования отчетности.

Предметом исследования выступает процесс проектирования и разработки информационной системы, адаптированной под специфику деятельности выбранного объекта, включая анализ, проектирование, реализацию и документирование.

Структура работы состоит из введения, трех глав, заключения, списка использованных источников и приложений. Первая глава (Аналитическая часть) посвящена теоретическим аспектам проектирования информационных систем, подробному анализу предметной области, выявлению проблем и задач, а также сравнению существующих решений (систем-аналогов). Вторая глава (Проектная часть) содержит этапы проектирования системы, обоснование выбора инструментов разработки, создание диаграмм (DFD, IDEF0, Use Case), проектирование базы данных и разработку макетов интерфейса. Третья глава

(Технологическая часть) описывает процесс программной реализации отдельных модулей системы, демонстрирует работу с базой данных, а также включает подготовленные руководства пользователя и программиста. Такое построение позволяет логично и последовательно раскрыть тему, от теоретического анализа до практической реализации.

1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1. Описание и анализ предметной области

Выпускная квалификационная работа охватывает широкий спектр объектов, процессов, понятий и правил, связанных с деятельностью отдела продаж салонов сотовой связи. Она рассматривается как часть реального мира, включающего процессы продажи мобильных устройств, аксессуаров, SIM-карт, подключения клиентов к тарифным планам и услугам операторов, а также управление информацией о клиентах, сотрудниках, товарах, поставках и контрактах. Эта область отличается высокой динамичностью, большим количеством участников и сложной структурой взаимосвязанных бизнес-процессов, что делает её значимой и сложной для автоматизации.

Салоны сотовой связи играют ключевую роль в продаже телекоммуникационных услуг и оборудования конечным потребителям. Они представляют собой розничные точки, где клиенты могут купить мобильные телефоны, планшеты, аксессуары и SIM-карты, а также подключиться к мобильной связи, изменить тарифный план, получить консультации и оформить рассрочку или кредит. В современных условиях салоны часто становятся центрами цифрового взаимодействия, предоставляя доступ к различным цифровым сервисам. Основная задача салона — удовлетворение потребностей клиентов в телекоммуникациях и оборудовании, при этом обеспечивая прибыльность и эффективность бизнеса.

Внутри предметной области выделяются основные участники, взаимодействующие с системой. Это клиенты, физические и юридические лица, инициирующие запросы на продажу или подключение товаров и услуг. Сотрудники, включая менеджеров и продавцов-консультантов, обслуживают клиентов, оформляют продажи и контракты, предоставляют консультации и выполняют планы. Старшие менеджеры и супервайзеры контролируют работу младшего персонала и управляют клиентами. Управляющие салоном и администраторы отвечают за планирование, контроль запасов и отчетность. Внутренние подразделения, такие как склад, бухгалтерия и отдел кадров,

также тесно взаимодействуют с отделом продаж. Поставщики, обеспечивающие товары и гарантийное обслуживание, являются внешними участниками.

Основные объекты предметной области включают различные элементы данных, используемые в бизнес-процессах. Это информация о клиентах, их персональные данные, контактные реквизиты и история взаимодействия. Важны данные о сотрудниках, включая ФИО, должность, логин и права доступа. Объектами являются товары с информацией о названии, типе, производителе, модели, цене и наличии. Данные о тарифных планах и услугах включают их названия, описания, стоимость и особенности. Ключевыми объектами являются записи о продажах и контрактах, фиксирующие сделки и юридически значимые соглашения. Заказы и заявки представляют собой запросы на товар или услугу от клиента.

Основные бизнес-процессы в отделе продаж салонов сотовой связи образуют логически связанную цепочку. Начинается с приема и обработки запросов клиентов, фиксируется интерес к товару или услуге. Затем следует этап консультации, где сотрудник предоставляет информацию о товарах и услугах, помогает подобрать оптимальное решение. Завершается оформление продажи или контракта, ввод данных о сделке, расчет стоимости, подписание документов и прием оплаты. Далее следует внутренний процесс комплектации и передачи товара клиенту, активации SIM-карт и настройки устройств. Важны также процессы управления складскими запасами, взаимодействия с поставщиками и формирования отчетности для анализа эффективности.

Традиционные методы учета, основанные на бумажной документации или электронных таблицах, сталкиваются с проблемами, снижающими эффективность работы отдела продаж. Низкая скорость обработки информации, ручной поиск, ввод и обновление данных требуют много времени. Высокий риск ошибок из-за человеческого фактора может привести к некорректным продажам или потере информации. Доступ к данным часто затруднен, они могут быть разрознены, недоступны в реальном времени или

храниться в неудобном формате, что осложняет анализ и отчетность. Сложность масштабирования и отсутствие контроля изменений затрудняют выявление ошибок и несанкционированного доступа. Взаимодействие между подразделениями может быть неэффективным из-за задержек или искажения информации.

Автоматизация процессов отдела продаж с помощью информационной системы решает эти проблемы и повышает эффективность работы. Централизованное и оперативное управление данными становится возможным благодаря структурированной базе данных. Точность и скорость обработки информации улучшаются за счет исключения ручного ввода и автоматизации рутинных операций. Качество обслуживания клиентов повышается благодаря быстрому поиску и предоставлению информации, точному расчету стоимости и выбору опций. Упрощается анализ и принятие решений — система формирует аналитические отчеты, сравнивает продажи, оценивает эффективность сотрудников и анализирует популярность товаров. Внедрение системы обеспечивает контроль и аудит действий пользователей, повышает прозрачность бизнес-процессов. В целом, автоматизация повышает эффективность салона, снижает издержки и повышает конкурентоспособность.

Анализ предметной области показывает, что современный салон сотовой связи — сложная система с множеством взаимодействующих элементов и процессов. Проблемы в системах учета и управления подтверждают актуальность и важность автоматизации.

Организационная структура управления салоном сотовой связи представлена на рисунке 1.1. Схема демонстрирует иерархию управления, распределение ответственности между подразделениями и должностями, что важно для определения пользователей будущей информационной системы и их ролей. Генеральный директор координирует работу всех отделов, включая маркетинг, финансы, кадры, ИТ, логистику, закупки, юридический отдел, бухгалтерию и директора салона N. Директор салона N управляет заместителем директора, который курирует менеджера по продажам,

уборщика, кассира и системного администратора. Менеджер по продажам руководит старшим продавцом-консультантом, который управляет продавцом-консультантом. Такая структура четко определяет функции пользователей информационной системы, что важно для её проектирования.

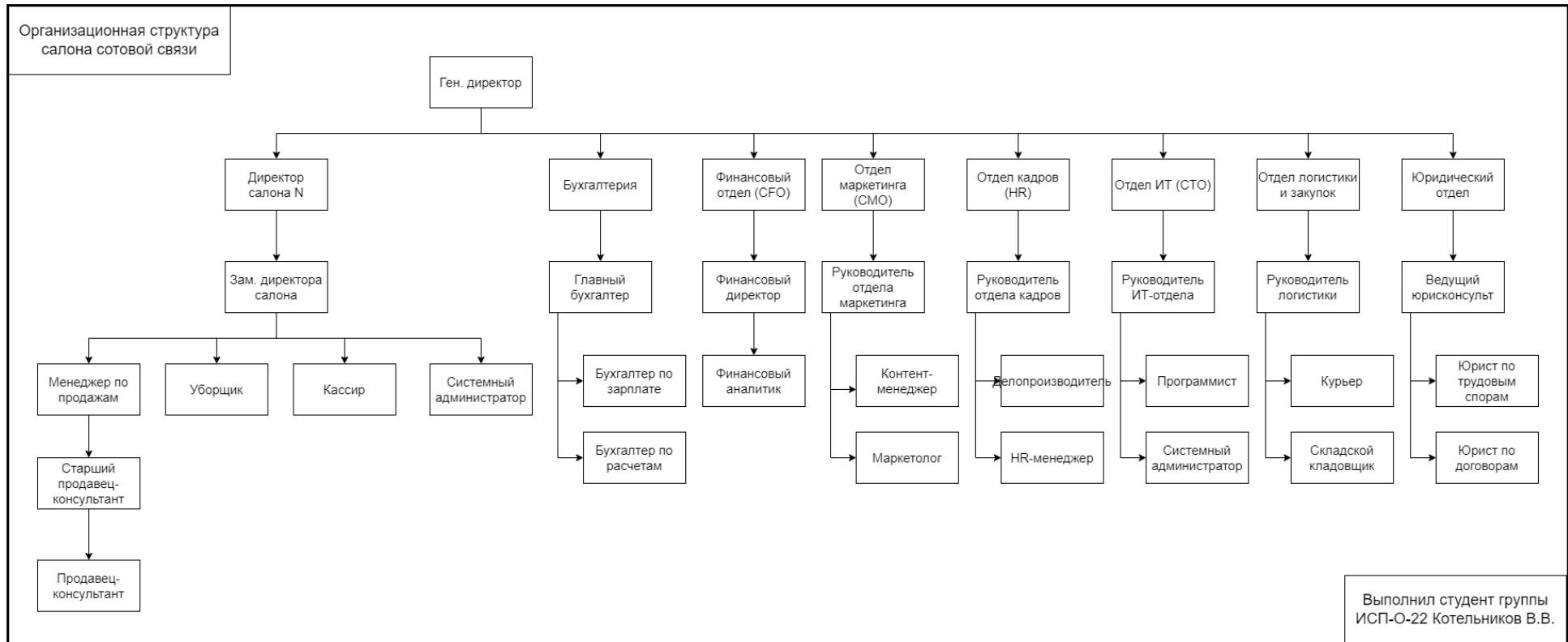


Рисунок 1.1 «Организационная структура»

1.2. Принципы разработки ИС

Разработка информационной системы (ИС) для салона сотовой связи — это сложный и многогранный процесс, требующий учета множества факторов. Основная цель — создать надежную, удобную, масштабируемую и экономически выгодную систему, определяющую архитектуру, функциональные возможности и методы работы на всех этапах разработки, тестирования и поддержки. В этом документе подробно изложены ключевые принципы, которые необходимо учитывать при создании ИС для салона сотовой связи.

1) Полнота функционала. ИС должна полностью соответствовать задачам автоматизации, выявленным при анализе предметной области. Это включает управление базой данных клиентов и сотрудников, каталогизацию товаров и тарифов, оформление и учет продаж, а также интеграцию с системами складского и бухгалтерского учета. Интерфейс системы должен быть интуитивно понятным для пользователей, таких как менеджеры и администраторы, что сократит время на обучение и уменьшит вероятность ошибок. Важно предусмотреть возможность адаптации системы под специфические нужды бизнеса, например, учет сезонных акций или интеграция с платежными системами.

2) Надежность и целостность данных. ИС должна обеспечивать высокую надежность хранения и обработки данных, защищая их от потерь, ошибок и несанкционированного доступа. Основные меры включают валидацию данных, контроль доступа, регулярное создание резервных копий и восстановление системы. Система должна быстро восстанавливаться после сбоев, сохраняя целостность и доступность данных. Важно обеспечить отказоустойчивость на уровне серверов и баз данных, а также предусмотреть механизмы защиты от атак, таких как DDoS.

3) Информационная безопасность. ИС будет работать с конфиденциальной информацией, включая личные данные клиентов и финансовые сведения. Обеспечение высокого уровня защиты данных является

ключевой задачей. Необходимо внедрить механизмы аутентификации и авторизации, шифрования данных, защиты от несанкционированного доступа, изменения и уничтожения. Регулярное резервное копирование и надежное восстановление данных играют важную роль. Важно учитывать требования законодательства в области защиты персональных данных, такие как GDPR.

4) Удобство интерфейса. Интерфейс ИС должен быть логично организован, интуитивно понятен и удобен для пользователей, что сократит время на обучение и уменьшит вероятность ошибок. Четкая структура меню, понятные обозначения полей и наличие справочной информации увеличат производительность. Необходимо предусмотреть адаптивный дизайн для мобильных устройств, чтобы пользователи могли работать с системой в любом месте и в любое время.

5) Высокая производительность. ИС должна оперативно реагировать на действия пользователей, быстро выполнять запросы и выдавать результаты. Это особенно важно при работе с большими объемами данных. Оптимизация архитектуры системы и использование передовых технологий обеспечат высокую производительность даже при росте числа пользователей или объемов данных. Важно предусмотреть механизмы кэширования и балансировки нагрузки для равномерного распределения ресурсов.

6) Масштабируемость системы. ИС должна быть разработана с учетом возможности легкого масштабирования, что важно для адаптации к расширению бизнеса, например, при открытии новых салонов или увеличении числа пользователей. Масштабируемость позволит системе сохранять эффективность и надежность даже в условиях роста. Важно предусмотреть модульную архитектуру, которая позволит добавлять новые модули и функции без значительных изменений существующей системы.

7) Интеграция с внешними системами. Современные предприятия активно используют различные программные продукты, такие как бухгалтерские системы, CRM и системы управления складом. ИС должна поддерживать интеграцию с этими системами через API или стандартные

протоколы обмена, обеспечивая бесшовную передачу данных и улучшая координацию между подразделениями. Важно предусмотреть возможность интеграции с внешними сервисами, такими как платежные системы или системы аналитики.

8) Автоматизация бизнес-процессов. Одной из главных целей внедрения ИС является автоматизация рутинных задач, таких как формирование заказов, расчеты, обновление складских остатков, генерация отчетов и уведомлений. Автоматизация уменьшает ручной труд, освобождая сотрудников для выполнения более сложных задач и повышая общую эффективность бизнеса. Важно предусмотреть возможность настройки автоматизации под конкретные бизнес-процессы салона.

9) Соответствие законодательным требованиям. При разработке ИС необходимо учитывать применимые законы и стандарты, такие как защита персональных данных, бухгалтерский учет и электронный документооборот. Система должна соответствовать международным и национальным стандартам проектирования ПО, таким как UML, и стандартам данных. Важно предусмотреть возможность аудита системы и предоставления отчетов для проверяющих органов.

10) Экономическая целесообразность. ИС должна не только выполнять свои функции, но и быть экономически выгодной. Это предполагает использование рациональных технологий, учет затрат на разработку и внедрение, обеспечение окупаемости инвестиций и возможность поэтапного внедрения. Экономическая целесообразность включает анализ затрат на поддержку и обслуживание системы, а также прогнозирование будущих расходов. Важно предусмотреть возможность модернизации системы без значительных затрат

1.3. Характеристика проблемы разработки

Разработка информационной системы для отдела продаж салонов сотовой связи — это многогранная задача, которая требует тщательного планирования и учета множества нюансов. Несмотря на очевидные

преимущества, такие как повышение эффективности работы, автоматизация процессов и улучшение взаимодействия с клиентами, этот проект сталкивается с рядом специфических вызовов. Понимание и грамотное решение этих проблем на этапе планирования и проектирования являются ключевыми факторами успешной реализации системы.

1) Сложность и разнообразие бизнес-процессов. Отдел продаж салона сотовой связи выполняет широкий спектр задач, включая не только продажу товаров, но и консультационные услуги, подбор тарифных планов, оформление контрактов, работу с рассрочкой и кредитами, а также обработку возвратов и гарантийное обслуживание. Каждый из этих процессов имеет свои особенности, условия и исключения, которые зависят от типа клиента, конкретного товара, действующего тарифа, текущих акций и политики салона. Моделирование и реализация подобных сложных и динамичных бизнес-процессов требуют глубокого анализа, четкого определения требований и разработки гибкой архитектуры, способной адаптироваться к любым изменениям.

2) Интеграция с внешними системами. Современные компании активно используют различные программные продукты, такие как системы бухгалтерского учета, управления складом, CRM-системы, учет персонала, платежные шлюзы, внутренние порталы и системы взаимодействия с поставщиками. Информационная система для отдела продаж должна быть интегрирована с этими внешними системами для обеспечения обмена данными. Например, необходимо синхронизировать информацию о наличии товаров на складе, данные о сотрудниках из системы управления персоналом и передавать данные о продажах в бухгалтерские системы. Организация надежного и безопасного обмена данными, согласование форматов и протоколов передачи информации — это сложная и ответственная задача, требующая тщательного подхода.

3) Обеспечение безопасности и конфиденциальности данных. Система будет обрабатывать персональные данные клиентов, включая ФИО, номера

телефонов, паспортные данные, историю покупок, а также данные сотрудников и финансовую информацию. Это требует строгого соблюдения законодательства, например, Федерального закона № 152-ФЗ в Российской Федерации, и внедрения надежных механизмов защиты информации. К таким механизмам относятся шифрование данных, аутентификация и авторизация пользователей, аудит действий, а также защита от несанкционированного доступа и кибератак. Нарушение безопасности может не только привести к юридическим последствиям, но и подорвать доверие клиентов к компании.

4) Разнообразие и специфика пользователей. Пользователями системы будут сотрудники с разными уровнями доступа и полномочиями, включая менеджеров, старших менеджеров и администраторов. Каждая роль предъявляет свои специфические требования к интерфейсу и функционалу системы. Интерфейс должен быть интуитивно понятным для пользователей с разным уровнем компьютерной грамотности, обеспечивать высокую скорость выполнения операций и минимизировать вероятность ошибок. Важным аспектом является также сопротивление изменениям со стороны персонала, которое может быть вызвано привычкой к старым методам работы. Поэтому при проектировании и внедрении системы необходимо уделить внимание обучению сотрудников и минимизации негативных последствий изменений.

5) Управление данными и обеспечение их целостности. В системе будет храниться большой объем взаимосвязанных данных, включая информацию о клиентах, сотрудниках, товарах, продажах и контрактах. Важно обеспечить целостность этих данных, чтобы избежать ошибок, таких как удаление товара, который уже участвовал в продаже, или создание контракта с несуществующим клиентом. Для этого необходимо разработать и внедрить правила валидации данных, ограничения на уровне базы данных и приложений, а также механизмы контроля и аудита действий пользователей. Это требует тщательной проработки и тестирования на всех этапах разработки.

6) Масштабируемость и производительность. Система должна быть спроектирована с учетом возможного роста бизнеса: увеличения числа

клиентов, объема продаж, количества салонов или ассортимента товаров. Архитектура и база данных должны обеспечивать стабильную работу даже при увеличении нагрузки. Снижение производительности может привести к тому, что система станет непригодной для использования, что негативно скажется на деятельности предприятия. Поэтому важно заранее предусмотреть меры по оптимизации производительности и масштабированию системы.

7) Неоднозначность и изменчивость требований. В процессе анализа и разработки системы могут выявляться новые аспекты бизнес-процессов или изменяться требования со стороны заказчика. Чтобы минимизировать риски, связанные с изменениями, необходимо четко документировать требования, постоянно взаимодействовать с заинтересованными сторонами и использовать гибкий подход к разработке. Это позволит оперативно реагировать на изменения и адаптировать систему к новым условиям.

8) Выбор и освоение инструментальных средств. Выбор языка программирования, фреймворков, системы управления базами данных (СУБД) и других технологий является важным этапом разработки. При этом необходимо учитывать не только функциональные требования, но и технические аспекты, такие как доступность ресурсов, навыки разработчиков, стоимость лицензий, а также перспективы развития технологий. Освоение новых или сложных инструментов требует времени и усилий, поэтому важно выбрать те, которые будут наиболее эффективными и удобными для команды разработчиков.

Разработка информационной системы для отдела продаж салонов сотовой связи представляет собой комплексную задачу, требующую учета множества факторов. Успешное преодоление всех перечисленных трудностей позволит создать эффективную, надежную, безопасную и удобную систему, которая станет мощным инструментом для автоматизации и оптимизации деятельности компании. Внедрение такой системы позволит повысить

качество обслуживания клиентов, улучшить взаимодействие с партнерами и повысить общую эффективность работы предприятия.

1.4. Обзор и сравнительный анализ готовых решений

Для обоснованного выбора архитектурного и технологического решения при разработке информационной системы отдела продаж салонов сотовой связи, 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.

Недостатки:

- Избыточность: для салона сотовой связи функционал может быть чрезмерно сложным и перегруженным.
- Высокая стоимость: Значительные затраты на лицензии, внедрение и сопровождение.
- Сложность внедрения: требуется длительное время и квалифицированные специалисты для настройки под конкретные нужды.

- Жёсткая архитектура: сложно вносить изменения или адаптировать под специфические требования.

Вывод: ERP-системы представляют собой мощные и универсальные решения, но для нужд конкретного отдела продаж салона сотовой связи они избыточны, дороги и сложны в настройке. Их внедрение оправдано, как правило, только на уровне крупного предприятия с высокой сложностью бизнес-процессов.

2) Специализированные решения (например, 1С:Розница, готовые CRM-системы)

Назначение: Программные продукты, ориентированные на конкретную сферу деятельности, например, розничная торговля или управление взаимоотношениями с клиентами.

Преимущества:

- Специализация: учитывают типичные бизнес-процессы в сфере (например, 1С:Розница – для розничной торговли).
- Готовая функциональность: включают в себя базовые функции, необходимые для работы (учёт товаров, продаж, клиентов).
- Интеграция с бухгалтерией: В случае с 1С – тесная интеграция с бухгалтерскими модулями.

Недостатки:

- Ограниченнная гибкость: сложно адаптировать под уникальные или нестандартные бизнес-процессы.
- Сложность интеграции с внешними сервисами: Подключение специфичных тарифов, систем лояльности или взаимодействие с внутренними системами оператора связи может быть затруднено.
- Зависимость от поставщика: Изменения и доработки зависят от разработчика.
- Стоимость: даже специализированные решения могут иметь высокую стоимость лицензирования и сопровождения.

Вывод: Специализированные решения, такие как 1С: Розница, могут покрыть значительную часть функционала отдела продаж. Однако, они могут не учитывать специфику работы с тарифами, контрактами, системами лояльности оператора связи, а также иметь ограниченные возможности для глубокой интеграции с внутренними системами компании. Это делает их не всегда идеальным выбором для салонов с нестандартными или высокодинамичными процессами.

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. Они предоставляют определённый функционал и гибкость. Однако, требуют высокой квалификации разработчиков для настройки,

интеграции и обеспечения безопасности. При отсутствии внутренних ИТ-ресурсов, это может стать проблемой.

4) SaaS-решения (например, Bitrix24, amoCRM, Р-Технологии)

Назначение: Программное обеспечение как услуга, предоставляемое через интернет на основе подписки.

Преимущества:

- Быстрое внедрение: не требует установки и настройки локальной инфраструктуры.
- Низкий порог входа: Простота начального запуска.
- Обновления: производятся автоматически провайдером.
- Масштабируемость: легко добавлять пользователей и функции.

Недостатки:

- Ограниченнная гибкость: Возможности кастомизации ограничены функционалом провайдера.
- Зависимость от провайдера: Работа системы, доступ к данным, политика безопасности зависят от внешней компании.
- 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 ИТ-ресурсов. Однако, при стремлении к полной автоматизации и контролю над всеми процессами, а также при работе с конфиденциальными данными, стандартные SaaS-продукты могут не обеспечить необходимой гибкости, интеграции и уровня безопасности.

Заключение по анализу:

Проведённый анализ показывает, что универсального "коробочного" решения, идеально подходящего под специфику деятельности отдела продаж салона сотовой связи, не существует. Стандартные ERP-системы избыточны и дороги. Специализированные решения решают отдельные задачи, но могут не

учитывать всю специфику работы с тарифами, контрактами и внутренними процессами, а также создают определённую фрагментацию и зависимость. Open Source и SaaS решения требуют либо высокой квалификации, либо компромиссов в функциональности, безопасности и контроле.

Таким образом, наиболее целесообразным представляется разработка собственной информационной системы, спроектированной с нуля под конкретные бизнес-процессы отдела продаж. Это позволит:

- Создать гибкое и масштабируемое решение, адаптированное под уникальные требования.
- Обеспечить полную интеграцию всех ключевых процессов (продажи, контракты, клиенты, сотрудники, склад).
- Учесть все специфические особенности работы с тарифами, акциями, системами лояльности.
- 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 данной выпускной квалификационной работы требования формулируются с учётом анализа предметной области, выявленных проблем и специфики деятельности салона сотовой связи.

Целью разрабатываемой информационной системы является автоматизация и оптимизация ключевых бизнес-процессов отдела продаж салонов сотовой связи. Это включает в себя:

- Повышение оперативности и точности обработки заказов и контрактов клиентов.
- Централизованное и структурированное хранение информации о клиентах, сотрудниках, товарах, тарифных планах.
- Упрощение и ускорение процесса поиска и предоставления информации.
- Автоматизацию рутинных операций, таких как оформление продаж, формирование отчётов.
- Обеспечение разграничения доступа к информации в зависимости от роли пользователя.
- Повышение точности учёта и снижение вероятности ошибок.
- Обеспечение аналитики по продажам, эффективности сотрудников и популярности товаров/тарифов.

Требования к ИС 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 отдела продаж салонов сотовой связи к функциональным требованиям относятся:

1) Управление клиентами (Client Management):

- F1.1. Возможность добавления, просмотра, редактирования и поиска информации о клиентах (ФИО, контактные данные, история покупок, подключенные услуги).
- F1.2. Возможность регистрации нового клиента в системе при оформлении первой продажи или контракта.
- F1.3. Возможность просмотра истории взаимодействия с конкретным клиентом (предыдущие продажи, контракты).

2) Управление сотрудниками (Employee Management):

- F2.1. Возможность добавления, просмотра, редактирования и поиска информации о сотрудниках (ФИО, должность, ID салона, данные для доступа).
- F2.2. Возможность управления учётными записями сотрудников (создание, изменение логина/пароля, сброс пароля, изменение уровня доступа/роли).

3) Управление товарами (Product Management):

- F3.1. Возможность добавления, просмотра, редактирования и поиска информации о товарах (название, тип, производитель, модель, цена, гарантия).
- F3.2. Возможность отображения информации о наличии товара (интеграция со складом или отображение актуальных остатков, если интеграция невозможна, то через ручное обновление).

4) Управление тарифными планами (Tariff Plan Management):

- F4.1. Возможность добавления, просмотра, редактирования и поиска информации о тарифных планах и услугах (название, описание, стоимость, особенности).

5) Управление продажами (Sale Management):

- F5.1. Возможность оформления новой продажи: выбор клиента, товара, ввод суммы, комментарии.

- F5.2. Возможность просмотра, поиска и фильтрации истории продаж.

- F5.3. Возможность (для определённых ролей) изменения или удаления продаж (например, при ошибках).

6) Управление контрактами (Contract Management):

- F6.1. Возможность оформления нового контракта: выбор клиента, тарифного плана, сотрудника, указание даты, статуса.

- F6.2. Возможность просмотра, поиска и фильтрации истории контрактов.

- F6.3. Возможность (для определённых ролей) изменения или удаления контрактов.

7) Система авторизации и ролевой модели (Authorization & Role Management):

- F7.1. Обязательная аутентификация пользователя (ввод логина и пароля) для доступа к системе.

- F7.2. Реализация ролевой модели с разными уровнями доступа (например, "Администратор", "Старший менеджер", "Менеджер"). Каждая роль имеет определённый набор разрешённых действий (CRUD-операции для разных сущностей).

Нефункциональные требования 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:

1) Производительность (Performance):

- NF1.1. Система должна отвечать на типичные запросы (просмотр списка клиентов, товаров, продаж) в течение 1-2 секунд при одновременной работе до 50 пользователей.

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

2) Надёжность (Reliability):

- 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).

3) Безопасность (Security):

- NF3.1. Пароли пользователей должны храниться в зашифрованном виде
- NF3.2. Реализация разграничения доступа к данным и функциям в зависимости от роли пользователя (например, менеджер не может удалять сотрудников, администратор может всё).

• NF3.3. Желательно ведение логов действий пользователей.

4) Удобство использования (Usability):

- NF4.1. Интерфейс должен быть интуитивно понятным, соответствовать общепринятым стандартам.
- NF4.2. Интерфейс должен быть локализован на русский язык.
- NF4.3. Система должна предоставлять пользователю понятные сообщения об ошибках или успешных действиях.

5) Совместимость (Compatibility):

- NF5.1. Система должна корректно работать в предполагаемой операционной системе (например, Windows 10/11).
- NF5.2. (Опционально) Возможность интеграции с бухгалтерской системой (например, 1С) или другими внутренними системами предприятия.

6) Масштабируемость (Scalability):

- 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 отдела продаж салонов сотовой связи. 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 (если он есть) и гарантирует, что финальный продукт будет соответствовать целям и задачам автоматизации, поставленным перед проектом.

2.2. Обзор и выбор средств проектирования и разработки ИС

Выбор инструментальных средств для разработки информационной системы отдела продаж салонов сотовой связи является критически важным этапом проекта. От правильности этого выбора зависит не только эффективность самого процесса разработки, но и конечные характеристики системы – её производительность, надежность, безопасность, удобство сопровождения и масштабируемости. В рамках данного проекта основной упор делается на технологии и инструменты, совместимые с операционной системой Windows, что соответствует предполагаемой среде эксплуатации. При выборе учитывались функциональные и нефункциональные требования, сформулированные ранее, доступность инструментов, наличие обширной документации и активного сообщества разработчиков.

Для реализации десктопного приложения, которое будет взаимодействовать с базой данных и предоставлять пользователям удобный графический интерфейс, логично обратиться к экосистеме Microsoft .NET. В этой среде ключевым языком программирования является C# (C-sharp). C# – это современный, объектно-ориентированный, строго типизированный язык, разработанный Microsoft. Он обладает рядом преимуществ: высокой производительностью, богатой стандартной библиотекой классов, мощными встроенными средствами для работы с базами данных, поддержкой асинхронного программирования и отличной интеграцией с операционной системой Windows. Эти характеристики делают C# идеальным выбором для создания надежных, масштабируемых и легко сопровождаемых настольных приложений на платформе .NET. Альтернативы, такие как VB.NET или Java, либо уступают в интеграции и удобстве разработки под Windows, либо требуют иных платформ и сред исполнения.

Для написания кода на C# и управления проектом необходима полнофункциональная интегрированная среда разработки (IDE). Наиболее подходящей и мощной IDE для разработки на .NET является Microsoft Visual Studio. Community-версия Visual Studio является бесплатной для студентов и индивидуальных разработчиков, что делает её привлекательной для проектов в рамках обучения. Visual Studio предоставляет исключительную поддержку языка C#: интеллектуальная система автодополнения (IntelliSense), встроенный отладчик, средства профилирования производительности, встроенные возможности управления системой контроля версий Git, а также визуальные конструкторы для создания интерфейсов Windows Forms или WPF. Это значительно ускоряет процесс разработки и упрощает отладку. Альтернативные IDE, такие как Visual Studio Code или JetBrains Rider, также обладают высоким потенциалом, но Visual Studio Community остается наиболее полной и интегрированной средой для разработки .NET Desktop приложений.

Для создания графического интерфейса (GUI) десктопного приложения в .NET доступны несколько технологий. В рамках проекта, ориентированного на функциональность и быструю разработку интерфейса для внутреннего использования, предпочтение отдается технологии Windows Forms (WinForms). WinForms – это зрелая и стабильная технология для создания настольных приложений с графическим интерфейсом в .NET Framework и .NET/.NET Core. Она предоставляет визуальный конструктор форм, богатую библиотеку элементов управления (controls) и относительно простой способ привязки данных. Для задачи автоматизации бизнес-процессов отдела продаж, где важна ясность и функциональность интерфейса, WinForms остаётся эффективным и быстрым в освоении решением. Альтернатива – WPF – более современна и гибка, но требует более глубоких знаний и времени на освоение.

Центральным элементом информационной системы является база данных, обеспечивающая хранение, управление и целостность всех данных. Для проекта, реализуемого в экосистеме Microsoft на платформе Windows,

естественным выбором становится СУБД Microsoft SQL Server. Это мощная, надежная, производительная реляционная система управления базами данных, обладающая развитыми средствами безопасности, масштабируемости и анализа. Особенno важно, что SQL Server обладает отличной интеграцией с .NET/C# через ADO.NET и Entity Framework, что упрощает разработку и повышает производительность. Для целей разработки и тестирования подходит бесплатная редакция SQL Server Express Edition, которая имеет достаточные ограничения (например, 10 ГБ на базу данных) для проектов среднего размера. Альтернативные open-source решения, такие как PostgreSQL или MySQL, также обладают высокими характеристиками, но SQL Server, учитывая интеграцию с .NET и доступность Express Edition, представляется наиболее подходящим выбором для данного проекта. SQLite, несмотря на свою легковесность, не подходит для многопользовательской среды салона связи.

Для визуального моделирования архитектуры системы, структуры базы данных, бизнес-процессов и взаимодействия с пользователями используются различные диаграммы. В проекте применяется инструмент draw.io (также известный как diagrams.net). Это бесплатный, веб- и десктопный инструмент с открытым исходным кодом, поддерживающий стандарты моделирования, такие как UML, BPMN, ER-моделирование (IDEF0, DFD, диаграммы Use Case). Он удобен, не требует сложной установки и легко интегрируется в документацию. Альтернативы включают коммерческие продукты, такие как Lucidchart или StarUML.

Наконец, важным аспектом разработки является контроль версий исходного кода. Для этих целей используется распределённая система контроля версий Git. Git позволяет эффективно отслеживать изменения в коде, работать с ветками, объединять изменения от разных разработчиков и создавать резервные копии проекта. Интеграция с Visual Studio делает его использование особенно удобным. Хостинг репозиториев, например, на

GitHub или GitLab, обеспечивает удалённое хранение кода и упрощает совместную работу.

Таким образом, в рамках проекта были выбраны следующие средства: язык программирования C#, среда разработки Visual Studio Community, технология интерфейса Windows Forms, СУБД Microsoft SQL Server (Express Edition), инструмент моделирования draw.io и система контроля версий Git. Такой набор технологий и инструментов позволяет эффективно реализовать все поставленные требования к информационной системе, обеспечивая высокое качество, надежность и удобство сопровождения конечного продукта.

2.3. Методы моделирования системы

Проектирование архитектуры приложения является ключевым этапом в разработке информационной системы. Оно определяет общую структуру программного обеспечения, взаимосвязи между его компонентами, а также принципы его функционирования и эволюции. Целью проектирования является создание надежной, гибкой, масштабируемой и легко сопровождаемой системы, удовлетворяющей предъявленным функциональным и нефункциональным требованиям. Архитектура служит планом для разработчиков, обеспечивая единое понимание структуры и поведения системы.

2.3.1. ER-диаграмма

ER-диаграмма (Entity-Relationship Diagram) — это графическая модель, используемая для представления структуры данных информационной системы. Она наглядно показывает основные сущности (объекты реального мира, например, "Клиент", "Продукт", "Заказ"), их атрибуты (характеристики сущностей, например, "Имя клиента", "Цена продукта") и связи между этими сущностями (например, "Клиент совершают Заказ"). Также на диаграмме указываются мощности (кардинальности) связей (один-к-одному, один-ко-многим, многие-ко-многим).

ER-диаграмма играет ключевую роль в этапе проектирования базы данных. Она позволяет:

1) Визуализировать структуру данных: четко представить, какие данные будут храниться в системе и как они связаны между собой.

2) Спроектировать базу данных: является основой для создания физической модели базы данных, то есть определения таблиц, их столбцов (атрибутов) и связей (внешних ключей) в выбранной СУБД.

3) Обеспечить согласованность: помогает разработчикам, аналитикам и заинтересованным сторонам выработать единое понимание структуры данных.

4) Выявить и устранить избыточность: позволяет проанализировать связи и атрибуты, способствуя созданию нормализованной и логически корректной структуры БД, тем самым минимизируя дублирование информации и потенциальные противоречия.

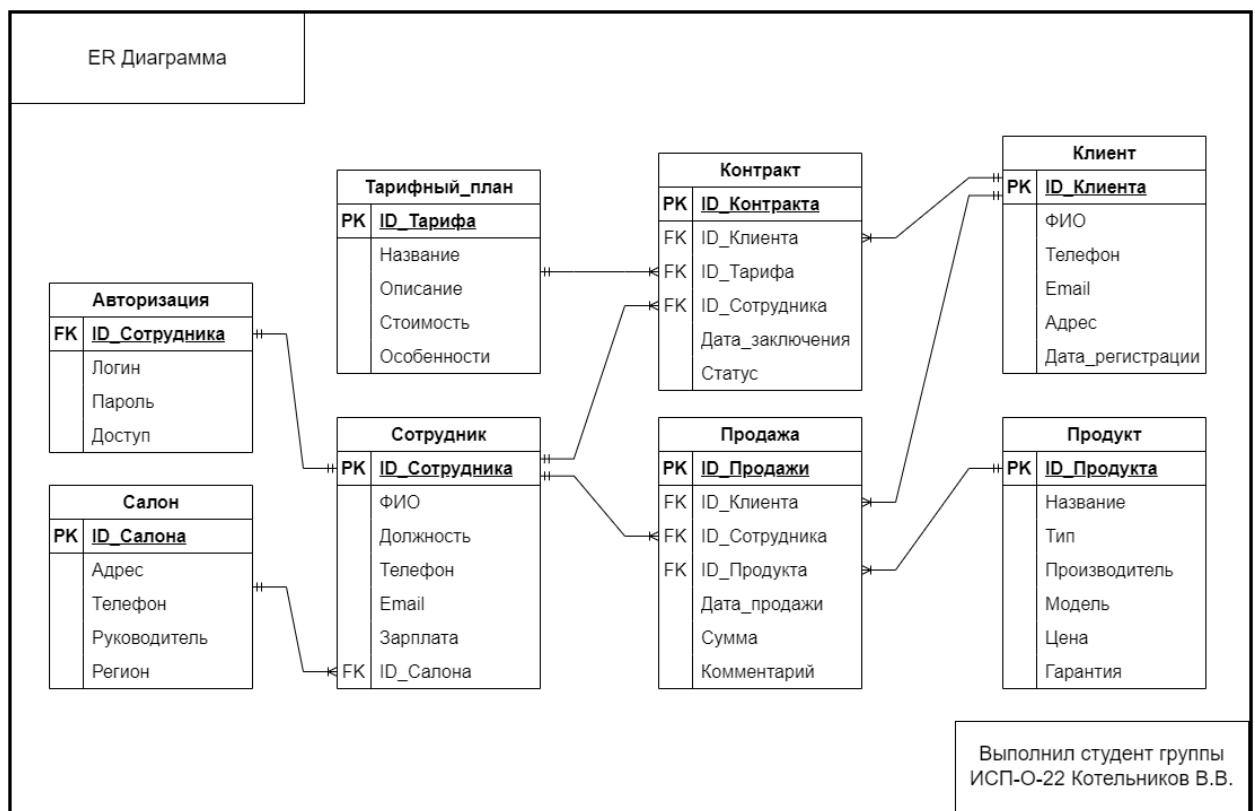


Рисунок 2.1. ER Диаграмма

На представленной ER-диаграмме визуализирована концептуальная структура базы данных информационной системы отдела продаж салонов сотовой связи. Диаграмма отражает ключевые сущности предметной области, их атрибуты и логические связи между ними, что служит основой для последующего проектирования физической модели базы данных.

Основные сущности (таблицы) и их атрибуты:

1) Сотрудник (Employee): Основная сущность, представляющая персонал салона. Идентифицируется уникальным ID_Сотрудника (первичный ключ). Содержит данные о сотруднике: ФИО, Должность, Телефон, Email, Зарплата. Связана с сущностью Салон через внешний ключ ID_Салона, что указывает на принадлежность сотрудника к конкретному салону.

2) Салон (Salon): Сущность, описывающая физическое или юридическое лицо, в котором работает сотрудник. Идентифицируется ID_Салона (первичный ключ). Атрибуты: Адрес, Телефон, Руководитель, Регион.

3) Авторизация (Auth): Сущность, хранящая учетные данные для входа в систему. Имеет первичный ключ ID_Auth и содержит Логин, Пароль, Доступ (уровень доступа). Связана с Сотрудником по ID_Сотрудника (внешний ключ), что означает, что каждый сотрудник имеет одну запись авторизации.

4) Клиент (Client): Сущность, представляющая покупателя или пользователя услуг. Идентифицируется ID_Клиента (первичный ключ). Атрибуты: ФИО, Телефон, Email, Адрес, Дата регистрации.

5) Продукт (Product): Сущность, описывающая товары, которые продаются в салоне. Идентифицируется ID_Продукта (первичный ключ). Атрибуты: Название, Тип, Производитель, Модель, Цена, Гарантия.

6) Тарифный план (TariffPlan): Сущность, описывающая услуги связи, предлагаемые операторами. Идентифицируется ID_Тарифа (первичный ключ). Атрибуты: Название, Описание, Стоимость, Особенности.

7) Продажа (Sale): Сущность, фиксирующая факт продажи товара клиенту. Идентифицируется ID_Продажи (первичный ключ). Атрибуты: Дата продажи, Сумма, Комментарий. Связана с Клиентом, Сотрудником и Продуктом через соответствующие внешние ключи (ID_Клиента, ID_Сотрудника, ID_Продукта).

8) Контракт (Contract): Сущность, фиксирующая заключение договора (например, подключение к тарифу). Идентифицируется ID_Контракта (первичный ключ). Атрибуты: Дата заключения, Статус. Связана с Клиентом,

Тарифным планом и Сотрудником через соответствующие внешние ключи (ID_Клиента, ID_Тарифа, ID_Сотрудника).

Связи между сущностями:

Все связи на диаграмме являются связями типа "один-ко-многим" (1:N), что отображается символами "одна линия" и "ветвь" (или "птичка"). Это означает, что один экземпляр сущности слева может быть связан с множеством экземпляров сущности справа.

- Салон — Сотрудник: Один салон может иметь много сотрудников.
- Сотрудник — Авторизация: Один сотрудник имеет одну запись авторизации (это частный случай 1:1, но технически реализуется как 1:N, где N=1).
- Сотрудник — Продажа / Контракт: Один сотрудник может совершить множество продаж и заключить множество контрактов.
- Клиент — Продажа / Контракт: Один клиент может совершить множество покупок и заключить множество контрактов.
- Продукт — Продажа: Один продукт может участвовать в множестве продаж (один и тот же телефон может быть продан разным клиентам).
- Тарифный план — Контракт: Один тарифный план может быть выбран множеством клиентов (заключено множество контрактов на один тариф).

2.3.2. Контекстная диаграмма

Контекстная диаграмма, соответствующая уровню A-0 в нотации IDEF0, представляет собой обобщённое описание системы как единой целостной функции и отражает её взаимодействие с окружающей средой. Она визуализирует центральный процесс системы (в виде одного прямоугольника), внешние субъекты (например, "Клиент", "Поставщик", "Сотрудник") и информационные потоки, соединяющие их с системой. Эти потоки подразделяются на:

- Входы (Input): Информация или ресурсы, поступающие в систему из внешней среды.

- Выходы (Output): Результаты функционирования системы, передаваемые во внешнюю среду.
- Управление (Control): Нормативные акты, правила, инструкции и прочее, регулирующие выполнение функций.
- Механизмы (Mechanism): Ресурсы, задействованные для выполнения функции (например, персонал, техника, программные средства).

Основная цель контекстной диаграммы — установить границы системы и понять, какие данные она принимает и отдает, а также какие внешние факторы оказывают на неё влияние.

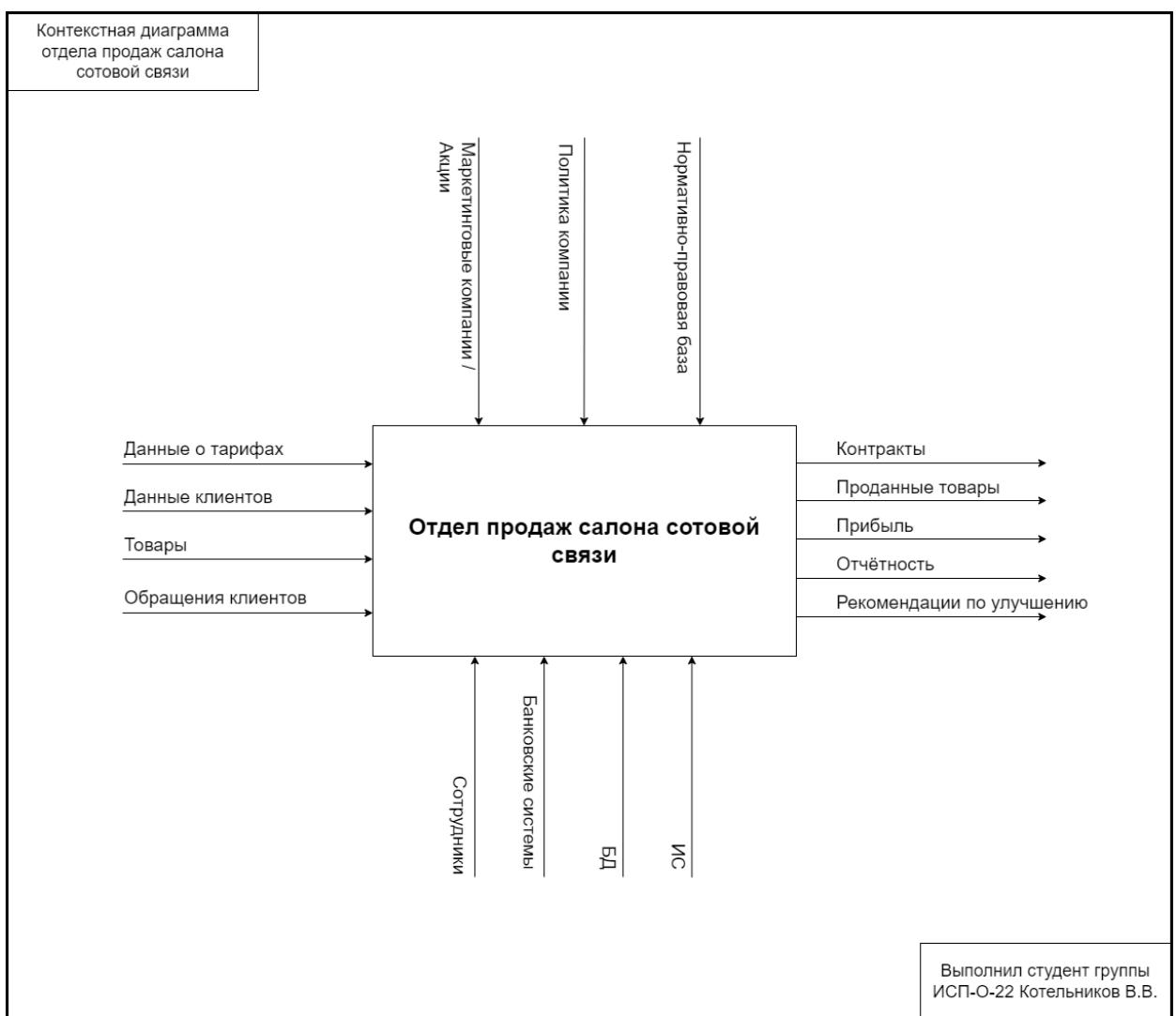


Рисунок 2.2. Контекстная диаграмма

На представленной контекстной диаграмме в нотации IDEF0 изображена информационная система "Отдел продаж салона сотовой связи" как единый процесс. Диаграмма определяет границы системы и показывает,

как она взаимодействует с внешней средой — другими подразделениями компании, внешними организациями и системами. Она служит отправной точкой для дальнейшего детального проектирования, отражая общую функцию системы: обработка данных, поступающих извне, и выдача результатов.

Центральный блок на диаграмме — "Отдел продаж салона сотовой связи" — представляет собой саму информационную систему. Вокруг этого блока расположены потоки данных, которые делятся на четыре категории согласно методологии IDEF0:

Входы (Input): это данные или ресурсы, поступающие в систему извне.

- Данные о тарифах — информация о доступных тарифных планах и услугах, необходимая для консультирования клиентов и оформления контрактов.
- Данные клиентов — информация о текущих и потенциальных клиентах, используемая для работы с базой данных и персонализированного обслуживания.
- Товары — данные о наличии, характеристиках и ценах товаров, предлагаемых в салоне.
- Обращения клиентов — запросы, жалобы, пожелания и заказы от клиентов, инициирующие бизнес-процессы внутри системы.

Выходы (Output): это результаты работы системы, передаваемые во внешнюю среду.

- Контракты — оформленные договоры, заключенные с клиентами.
- Проданные товары — информация о совершенных продажах, передаваемая, например, складу для списания и бухгалтерии для учета.
- Прибыль — финансовый результат деятельности отдела, рассчитываемый системой.
- Отчётность — аналитические и управленческие отчеты, предоставляемые руководству и другим подразделениям.

- Рекомендации по улучшению — выводы и предложения, сформированные на основе анализа данных, направленные на повышение эффективности работы.

Управление (Control): это правила, регламенты, политики и нормативно-правовая база, которые определяют, как система должна работать.

- Маркетинговые кампании / Акции — условия и правила проведения рекламных акций, влияющие на ценообразование и оформление продаж.
- Политика компании — внутренние стандарты и процедуры, которым должен следовать отдел продаж.
- Нормативно-правовая база — законы и регламенты, регулирующие деятельность салона сотовой связи (например, Закон "О защите прав потребителей", налоговое законодательство).

Механизмы (Mechanism): это ресурсы, задействованные для выполнения функций системы.

- Сотрудники — персонал салона, непосредственно работающий с клиентами и системой.
- Банковские системы — внешние платежные шлюзы и банковские сервисы, используемые для приема оплаты.
- БД — база данных, хранящая все оперативные данные системы.
- ИС — разрабатываемая ИС.

2.3.3. Диаграмма декомпозиции IDEF0

Диаграмма декомпозиции IDEF0 детализирует функции системы, разбивая общую задачу на более мелкие подпроцессы. Каждый подпроцесс описывается через четыре компонента: Вход, Выход, Управление и Механизмы. Это позволяет поэтапно, с нарастающей детализацией, раскрыть внутреннюю логику работы системы. Диаграммы связаны иерархически, обеспечивая согласованность на всех уровнях.

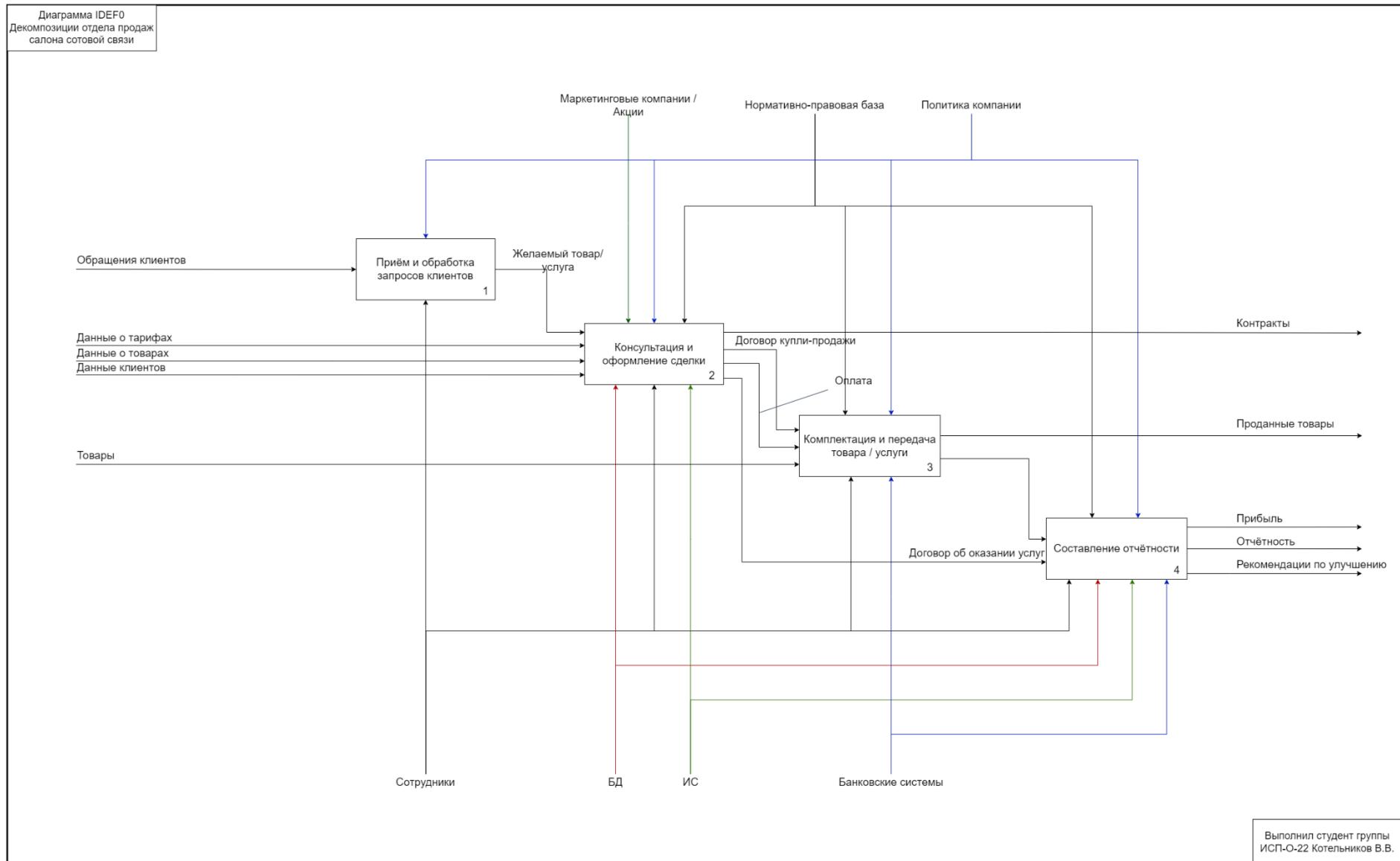


Рисунок 2.3. Диаграмма декомпозиции IDEF0

На представленной диаграмме декомпозиции IDEF0 (уровень A0) детализирована функция "Отдел продаж салона сотовой связи", показанная на контекстной диаграмме. Диаграмма разбивает эту единую функцию на четыре основных подпроцесса, каждый из которых описывается в соответствии с нотацией IDEF0: через входы (Input), выходы (Output), управление (Control) и механизмы (Mechanism). Это позволяет наглядно представить внутреннюю логику работы отдела продаж как системы.

Процессы и их взаимодействие:

1) Процесс 1: Приём и обработка запросов клиентов.

Входы: Обращения клиентов — основной импульс для начала работы.

Выходы: Желаемый товар/услуга — результат первичного анализа запроса клиента.

Механизмы: Сотрудники — персонал, осуществляющий прием и консультацию.

Управление: Маркетинговые компании / Акции, Нормативно-правовая база, Политика компании — правила и условия, которые должны учитываться при работе с клиентом.

2) Процесс 2: Консультация и оформление сделки.

Входы: Получает Желаемый товар/услуга от процесса 1, а также Данные о тарифах, Данные о товарах, Данные клиентов — необходимая информация для подбора решения и оформления.

Выходы: Договор купли-продажи — юридически значимый документ, фиксирующий сделку.

Механизмы: Сотрудники, БД (база данных), ИС (информационная система) — ресурсы, используемые для консультирования и оформления документов.

Управление: Маркетинговые компании / Акции, Нормативно-правовая база, Политика компании — правила, определяющие условия сделки.

3) Процесс 3: Комплектация и передача товара / услуги.

Входы: Получает Договор купли-продажи от процесса 2, а также Товары — физические или цифровые товары, которые нужно передать.

Выходы: Проданные товары — физический результат деятельности (товар передан клиенту).

Механизмы: Сотрудники, БД, ИС, Банковские системы — ресурсы, задействованные в выдаче товара и получении оплаты.

Управление: Нормативно-правовая база, Политика компании — правила, регулирующие выдачу товара и проведение платежей.

4) Процесс 4: Составление отчёtnости.

Входы: Получает Договор купли-продажи и Договор об оказании услуг от процессов 2 и 3. Также используется БД и ИС для сбора данных.

Выходы: Прибыль, Отчётность, Рекомендации по улучшению — ключевые результаты для управления бизнесом.

Механизмы: БД, ИС — системы, в которых хранятся данные для формирования отчётов.

Управление: Нормативно-правовая база, Политика компании — требования к составлению отчетов и анализу данных.

Взаимосвязи между процессами:

- Процесс 1 передает Желаемый товар/услуга в Процесс 2.
- Процесс 2 передает Договор купли-продажи в Процесс 3 и Процесс 4.
- Процесс 3 передает Проданные товары в Процесс 4.

Все процессы используют общие механизмы (Сотрудники, БД, ИС) и управляются едиными правилами (Маркетинговые компании / Акции, Нормативно-правовая база, Политика компании).

Эта диаграмма декомпозиции является важным инструментом для проектирования разрабатываемой информационной системы. Она четко определяет границы каждого бизнес-процесса, его входы и выходы, что напрямую влияет на проектирование модулей ИС. Например, процесс "Консультация и оформление сделки" будет реализован как модуль, работающий с данными о клиентах, товарах и тарифах, и генерирующий

документы. Процесс "Составление отчётности" потребует создания модуля аналитики, который будет использовать данные из других модулей. Диаграмма обеспечивает полное и согласованное понимание логики работы всей системы, что является основой для дальнейшей разработки программного кода и базы данных.

2.3.4. Диаграмма потоков данных

DFD-диаграмма — это графическая модель, которая визуализирует потоки информации и данных внутри организационной структуры. В отличие от диаграммы, описывающей только программное обеспечение, DFD в данном случае показывает, как данные перемещаются между различными подразделениями, должностями и внешними партнерами компании, а также где эти данные хранятся.

Её основная цель — понять, как информация проходит через бизнес-процессы организации. Это помогает выявить:

- Источники данных: Кто или что генерирует информацию? (например, клиенты, поставщики, отдел маркетинга).
- Потребители данных: Кто использует эту информацию? (например, бухгалтерия, отдел продаж, руководство).
- Места хранения данных: где информация временно или постоянно хранится? (например, файловые архивы, базы данных, электронные таблицы).
- Ключевые процессы: Какие действия выполняются над данными? (например, "Обработка заказа", "Формирование отчёта", "Согласование сметы").

Основные элементы DFD-диаграммы для организации:

1) Внешние сущности (Terminators): представляют собой акторов или системы, находящиеся вне моделируемой части организации, но взаимодействующие с ней.

Примеры: Клиент, Поставщик, Банковская система, Отдел кадров (если он не входит в границы моделируемого процесса).

2) Процессы (Processes): изображаются прямоугольниками с закругленными углами. Они представляют собой конкретные действия или функции, которые выполняются в организации, преобразуя входящую информацию в исходящую.

Примеры: Приём и обработка заявок, Оформление договора, Подготовка финансовой отчётности.

3) Хранилища данных (Data Stores): изображаются параллельными линиями. Они обозначают места, где информация накапливается и хранится для последующего использования.

Примеры: Архив договоров, База данных клиентов, Электронный реестр товаров, Журнал учета оплат.

4) Потоки данных (Data Flows): Линии со стрелками, показывающие направление движения информации. На каждой стрелке указывается, что именно передается (например, Заявка клиента, Расчетный документ, Данные о наличии товара).

Значение DFD-диаграммы для организации:

- Понимание бизнес-процессов: наглядно показывает, как информация движется по организации, что позволяет лучше понять логику работы.
- Выявление узких мест и дублирования: помогает найти участки, где информация задерживается, теряется или обрабатывается несколько раз.
- Улучшение коммуникации: создает единое представление о том, кто и что делает, что способствует более эффективному взаимодействию между подразделениями.
- Основа для автоматизации: является отправной точкой для разработки информационных систем, так как четко определяет, какие данные нужно собирать, обрабатывать и хранить.
- Оптимизация работы: позволяет выявить возможности для упрощения процессов, сокращения времени на выполнение задач и повышения качества предоставляемых услуг.

Таким образом, DFD-диаграмма — это мощный инструмент для анализа и проектирования деятельности организации, позволяющий визуализировать и оптимизировать потоки данных, что напрямую влияет на эффективность бизнеса.

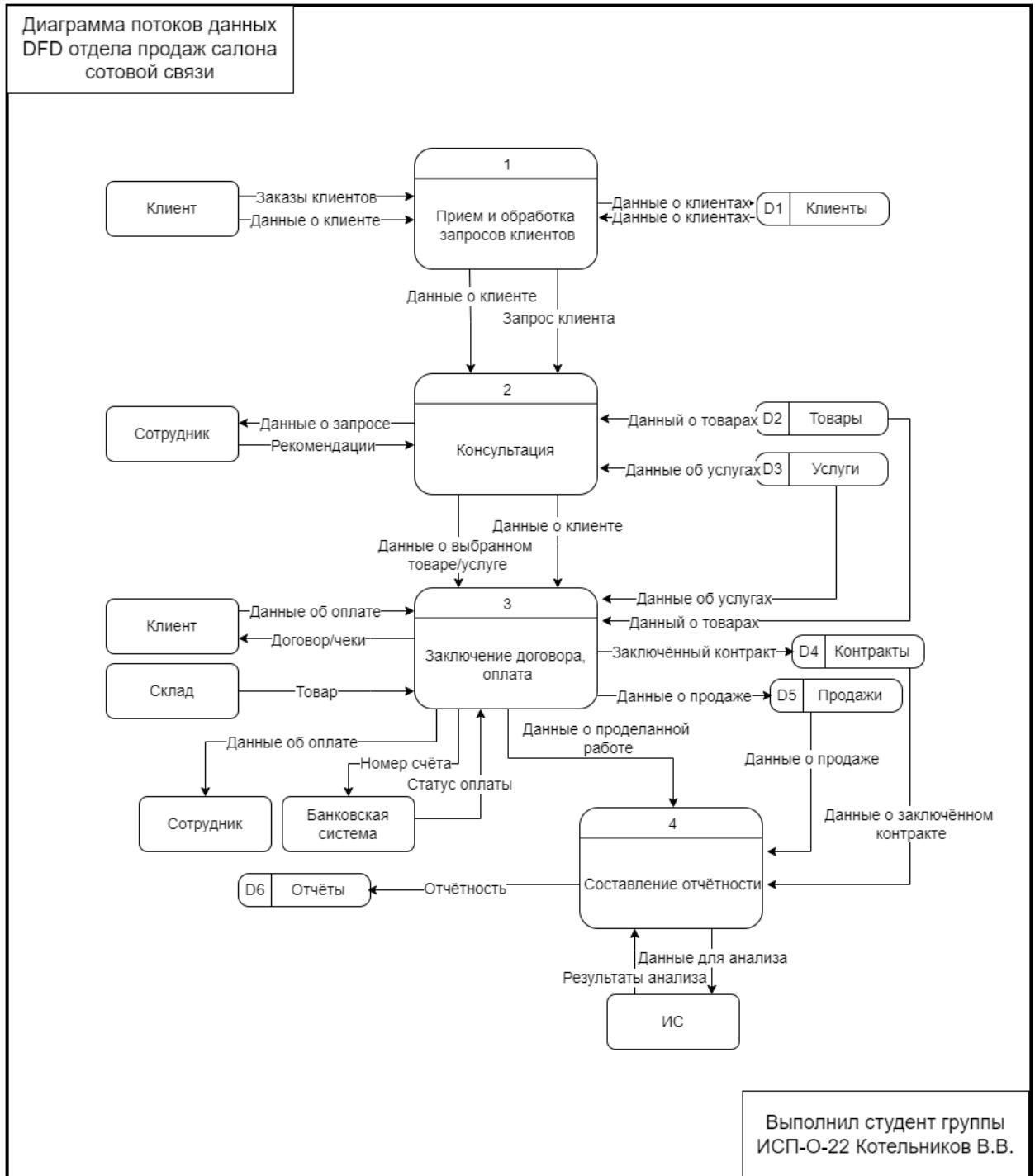


Рисунок 2.4. Диаграмма потоков данных

На представленной диаграмме потоков данных (DFD), выполненной в нотации Гейна-Сарсона, визуализирована логика обработки информации и взаимодействия между различными участниками (акторами) и процессами в

рамках деятельности отдела продаж салона сотовой связи. Эта диаграмма представляет собой логическую модель, которая описывает, как данные перемещаются по системе, какие процессы над ними выполняются, где они хранятся и с кем взаимодействует организация. Она является важным инструментом для анализа бизнес-процессов и служит основой для проектирования информационной системы.

Основные элементы диаграммы:

1) Внешние сущности (Terminators): представляют собой акторов или системы, находящиеся вне границ моделируемого процесса, но активно участвующих в обмене данными.

- Клиент: инициирует процесс, направляя Заказы клиентов и предоставляя Данные о клиенте. Получает Договор/чеки и Рекомендации.

- Сотрудник: участвует в консультировании (Данные о запросе) и получает Рекомендации, а также участвует в оформлении оплаты.

- Склад: Поставщик физического товара, передающий Товар в процесс оформления сделки.

- Банковская система: Внешняя система, осуществляющая финансовую операцию (Данные об оплате) и подтверждающая её (Номер счёта, Статус оплаты).

2) Процессы (Processes): Изображены прямоугольниками с закругленными углами. Каждый процесс — это действие, которое преобразует входящие данные в исходящие.

- Процесс 1: Прием и обработка запросов клиентов. Это начальный этап, где регистрируется Запрос клиента и собираются Данные о клиенте. Информация о клиенте записывается в хранилище D1 Клиенты.

- Процесс 2: Консультация. На этом этапе сотрудник работает с клиентом, используя данные из хранилищ D2 Товары и D3 Услуги для подбора решения. Результат — Данные о выбранном товаре/услуге.

- Процесс 3: Заключение договора, оплата. Ключевой процесс, который формирует юридически значимый документ (Заключённый контракт,

записываемый в D4 Контракты) и организует финансовые операции. Он взаимодействует с Складом для получения товара и с Банковской системой для проведения оплаты. Данные о совершенной продаже (Данные о продаже) записываются в D5 Продажи.

- Процесс 4: Составление отчётности. Финальный процесс, который собирает данные со всех предыдущих этапов (Данные о продаже, Данные о заключённом контракте) и генерирует Отчётность (записывается в D6 Отчёты). Этот процесс также может использовать ИС (информационную систему) для получения Данных для анализа и выдачи Результатов анализа.

3) Хранилища данных (Data Stores): Изображены параллельными линиями. Они представляют собой места, где информация временно или постоянно хранится.

- D1 Клиенты: хранит все данные о клиентах.
- D2 Товары: хранит информацию о номенклатуре, ценах и наличии товаров.
- D3 Услуги: хранит информацию о тарифных планах и услугах.
- D4 Контракты: хранит оформленные договоры.
- D5 Продажи: хранит данные о совершенных продажах.
- D6 Отчёты: хранит сформированные отчеты.

4) Потоки данных (Data Flows): Линии со стрелками, показывающие направление движения информации. На каждой стрелке указано, какой именно поток данных передается (например, Заказы клиентов, Данные о товаре, Отчётность).

Логика работы системы: Работа системы начинается с Клиента, который направляет Заказы клиентов в процесс 1. Данные о клиенте поступают в хранилище D1 Клиенты. Затем информация передается в процесс 2, где Сотрудник проводит Консультацию, используя данные из хранилищ D2 Товары и D3 Услуги. После выбора товара/услуги процесс 3 Заключение договора, оплата формирует Заключённый контракт (записывается в D4 Контракты) и взаимодействует с Складом и Банковской системой для

завершения сделки. Данные о продаже записываются в D5 Продажи. Наконец, процесс 4 Составление отчётности собирает данные из всех источников и генерирует Отчётность, которая может быть использована для анализа и принятия управленческих решений.

Значение диаграммы: Эта DFD-диаграмма служит фундаментом для понимания и оптимизации деятельности отдела продаж. Она наглядно демонстрирует, как информация циркулирует между различными подразделениями (Клиент, Сотрудник, Склад, Банковская система) и внутренними процессами. Это позволяет выявить узкие места, дублирование функций, а также определить, какие данные необходимо собирать, обрабатывать и хранить. Диаграмма является ключевым артефактом для дальнейшего проектирования информационной системы, так как четко определяет её функциональные блоки, потоки данных и точки взаимодействия с внешним миром.

2.3.5. Диаграмма вариантов использования (Use Case Diagram)

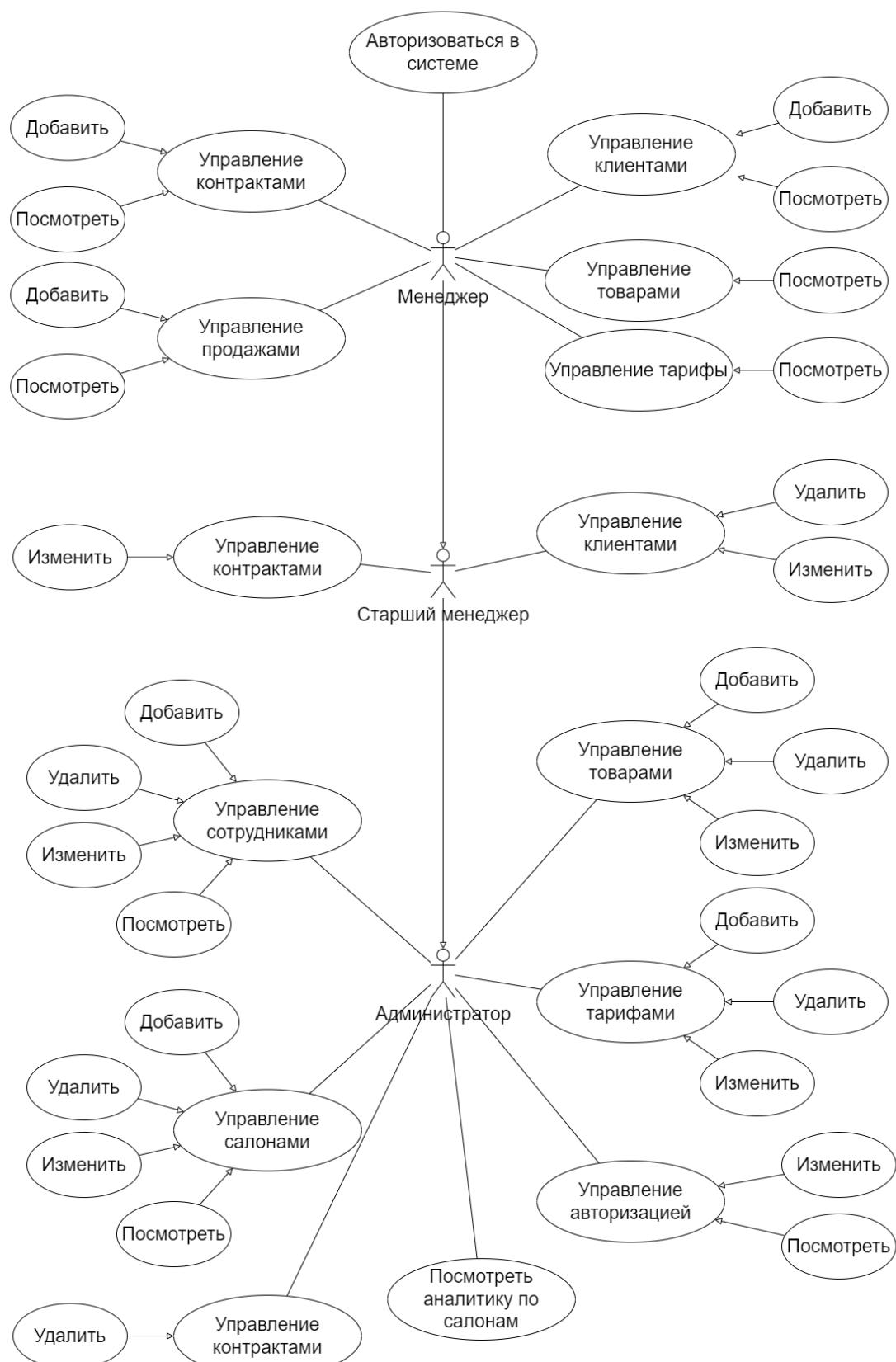
Use Case диаграмма — это один из типов диаграмм в языке унифицированного моделирования (UML), который используется для описания функциональности системы с точки зрения её пользователей. Она показывает, какие функции система предоставляет и кто их использует. Под актором понимается роль пользователя (например, "Менеджер", "Администратор") или внешняя система, взаимодействующая с разрабатываемой информационной системой (ИС). Вариант использования (Use Case) описывает определённое действие или цель, которую пользователь этой роли стремится достичь в рамках системы (например, "Создать заказ", "Просмотреть отчёт"). Диаграмма визуализирует, какие возможности (варианты использования) доступны для каждой роли (актора) в контексте, разрабатываемой ИС.

Основные элементы диаграммы:

- Акторы (Actor): обозначаются стилизованным человечком и представляют собой роли пользователей или другие системы, которые будут взаимодействовать с ИС.
- Use Case (Вариант использования): представляются в виде эллипсов и описывают конкретные функции или возможности, предоставляемые системой.
- Отношения: Сплошные линии (ассоциации) соединяют акторов с теми вариантами использования, которые им доступны. Также могут использоваться специальные типы отношений (include, extend) между самими вариантами использования для описания сложной логики.

Цель Use Case диаграммы — определить и визуализировать функциональные требования к системе с позиции пользователя, чётко обозначив, кто и какие возможности может использовать.

Диаграмма Use-Case отдела продаж салонов сотовой связи



Выполнил студент группы
ИСП-О-22 Котельников В.В.

Рисунок 2.5. Диаграмма вариантов использования

На представленной диаграмме вариантов использования (Use Case) в нотации UML отображены функциональные возможности разрабатываемой информационной системы отдела продаж салонов сотовой связи, сгруппированные по ролям пользователей. Диаграмма наглядно демонстрирует, какие действия (варианты использования) доступны каждой роли (актору) и как эти роли связаны между собой через отношение обобщения.

Основные элементы диаграммы:

1) Акторы (Actors): Изображены в виде стилизованных человечков и представляют собой роли пользователей системы.

- Менеджер: Базовая роль, имеющая доступ к основным операциям по работе с клиентами, товарами, тарифами, контрактами и продажами.
- Старший менеджер: Расширенная роль, которая наследует все права Менеджера и получает дополнительные полномочия.
- Администратор: Наивысшая роль, которая наследует все права Старшего менеджера и имеет полный контроль над системой, включая управление пользователями и аналитикой.

2) Варианты использования (Use Cases): Изображены в виде эллипсов и описывают конкретные цели или функции, которые пользователь может выполнить в системе. Каждый вариант использования может быть раскрыт на более детальном уровне (например, "Управление клиентами" включает подфункции "Добавить", "Посмотреть").

- Основные группы функций: Управление клиентами, Управление товарами, Управление тарифами, Управление контрактами, Управление продажами.
- Дополнительные функции для старших ролей: Управление сотрудниками, Управление салонами, Управление авторизацией, посмотреть аналитику по салонам.

3) Отношения:

Ассоциация: Сплошные линии, соединяющие акторов с их вариантами использования, показывают, какие функции доступны каждой роли.

Обобщение (Generalization): Отношение, изображенное сплошной линией с полой треугольной стрелкой, направленной от специализированного актора к общему. В данном случае:

- Старший менеджер является обобщением Менеджера. Это означает, что Старший менеджер автоматически наследует все функции Менеджера (например, просмотр и добавление данных) и получает дополнительные возможности (например, изменение данных).
- Администратор является обобщением Старшего менеджера. Следовательно, Администратор наследует все функции Старшего менеджера (включая все функции Менеджера) и получает уникальные полномочия, такие как управление учетными записями (Управление авторизацией) и просмотр аналитики по всем салонам.

Логика работы и назначение диаграммы:

Диаграмма четко определяет границы функциональности для каждой роли пользователя. Она показывает, что:

- Менеджер может выполнять базовые операции: просматривать данные и добавлять новые записи во все основные справочники (клиенты, товары, тарифы, контракты, продажи).
- Старший менеджер, помимо всех функций Менеджера, получает возможность изменять существующие записи в этих же справочниках, а также управлять данными о сотрудниках.
- Администратор обладает максимальными правами: он может не только изменять и просматривать все данные, но и удалять их, а также управлять самими пользователями системы и анализировать работу по всем салонам.

Таким образом, диаграмма служит фундаментом для реализации системы управления доступом (разграничения прав). Она позволяет точно определить, кто и какие действия может совершать в системе, что критически

важно для обеспечения безопасности, целостности данных и эффективного управления бизнес-процессами.

2.4. Макеты интерфейса приложения

Разработка пользовательского интерфейса (UI) является одной из ключевых задач на этапе проектирования информационной системы. Интерфейс служит посредником между пользователем и логикой приложения, и его качество напрямую влияет на эффективность работы, удобство использования и общее восприятие системы. Непосредственно перед программной реализацией важно создать чёткое визуальное представление того, как будет выглядеть и функционировать интерфейс. Для этой цели используются макеты интерфейса (mockups или wireframes).

Макеты интерфейса — это схематические или приближенные к финальному виду изображения окон, страниц или отдельных элементов приложения. Они позволяют визуализировать структуру экранов, расположение элементов управления (кнопок, полей ввода, таблиц и т.д.), навигацию между окнами и общую логику взаимодействия пользователя с системой. Создание макетов на этапе проектирования позволяет:

- Уточнить и согласовать требования: Наглядное изображение интерфейса помогает разработчикам и заинтересованным сторонам (заказчикам) лучше понять, как будет устроена система, и вовремя внести корректировки в функциональность.
- Выявить потенциальные проблемы юзабилити: на стадии макета можно проанализировать, насколько интуитивно понятен интерфейс, удобно ли пользователю находить нужные функции и вводить данные.
- Определить структуру и логику работы приложения: Макеты помогают чётко обозначить, какие действия пользователь может совершить на каждом экране, какая информация будет отображаться и как она будет обновляться.

- Сократить время и стоимость разработки: четко проработанные макеты служат точной инструкцией для программиста при реализации UI, уменьшая количество итераций и доработок.

В рамках данной выпускной квалификационной работы макеты интерфейса разрабатываются с учётом ролевой модели, определённой в диаграмме Use Case (см. п. 2.3), и требований к удобству использования. Они отражают основные сценарии взаимодействия для различных ролей пользователей: менеджера, старшего менеджера и администратора. Ниже представлены и описаны ключевые макеты основных окон разрабатываемой информационной системы отдела продаж салонов сотовой связи. Эти макеты послужат основой для программной реализации интерфейса в главе 3.

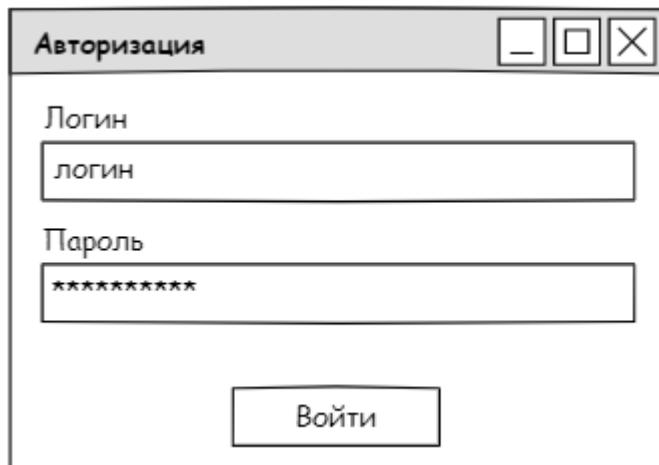


Рисунок 2.6. Макет окна авторизации

На представленном макете изображено стартовое окно информационной системы, предназначенное для аутентификации пользователя. Это первое окно, с которым взаимодействует пользователь при запуске программы.

Цель окна: проверить личность пользователя и предоставить доступ к системе на основе введенных учетных данных (логина и пароля).

Окно имеет стандартную заголовочную панель с названием "Авторизация" и кнопками управления (свернуть, развернуть, закрыть).

Внутри окна расположены два поля ввода:

- Логин: Текстовое поле, в которое пользователь вводит свой уникальный идентификатор (логин). В данном макете в качестве примера указано значение "логин".
- Пароль: Текстовое поле, в которое пользователь вводит свой пароль. Для обеспечения безопасности введенные символы отображаются звездочками ("*****"), скрывая реальное значение.

Под полями ввода расположена одна основная кнопка:

- Войти: Кнопка, по нажатию которой система проверяет введенные логин и пароль. При успешной аутентификации пользователь получает доступ к главному окну системы, соответствующему его роли (Менеджер, Старший менеджер, Администратор). В случае неверных данных система должна выдать сообщение об ошибке.

Этот макет отражает базовый и интуитивно понятный интерфейс входа, который является обязательным элементом любой многопользовательской системы для обеспечения безопасности и разграничения доступа.



Рисунок 2.7 Макет окна администратора

На представленном макете изображено главное окно для пользователя с ролью Администратора. Это центральная точка управления системой, предоставляющая администратору доступ к ключевым функциям: анализу данных по конкретному салону и управлению справочниками.

Цель окна: обеспечить администратору возможность проводить аналитику продаж по выбранному салону и управлять данными системы через соответствующие модули.

Окно имеет стандартную заголовочную панель с названием "Администрация" и кнопками управления (свернуть, развернуть, закрыть).

В верхней части окна расположена секция для проведения аналитики:

- Элемент управления: Выпадающий список (ComboBox) с предварительно загруженным списком всех салонов связи (например, "салон 1", "салон 2"). Пользователь выбирает нужный салон.
- Кнопка: "Анализ". При нажатии на эту кнопку система выполняет запрос к базе данных, рассчитывает показатели продаж за текущий месяц для выбранного салона и сравнивает их с данными за предыдущий месяц.
- Результат анализа (текстовое описание html) отображается в основной области окна.

Основная область: Большое поле, в котором отображаются результаты анализа.

В нижней части окна расположены две основные кнопки для навигации:

- Кнопка "Панель продаж": при нажатии открывает окно продаж, которое будет описано позже.
- Кнопка "Редактор данных": при нажатии открывает окно, предназначенное для управления справочниками системы. Администратор может здесь добавлять, изменять и удалять данные о клиентах, сотрудниках, товарах, тарифных планах и других сущностях.

Этот макет обеспечивает администратору удобный и логичный интерфейс для выполнения его основных задач: получения оперативной

аналитической информации по работе салонов и управления всей информационной базой системы.

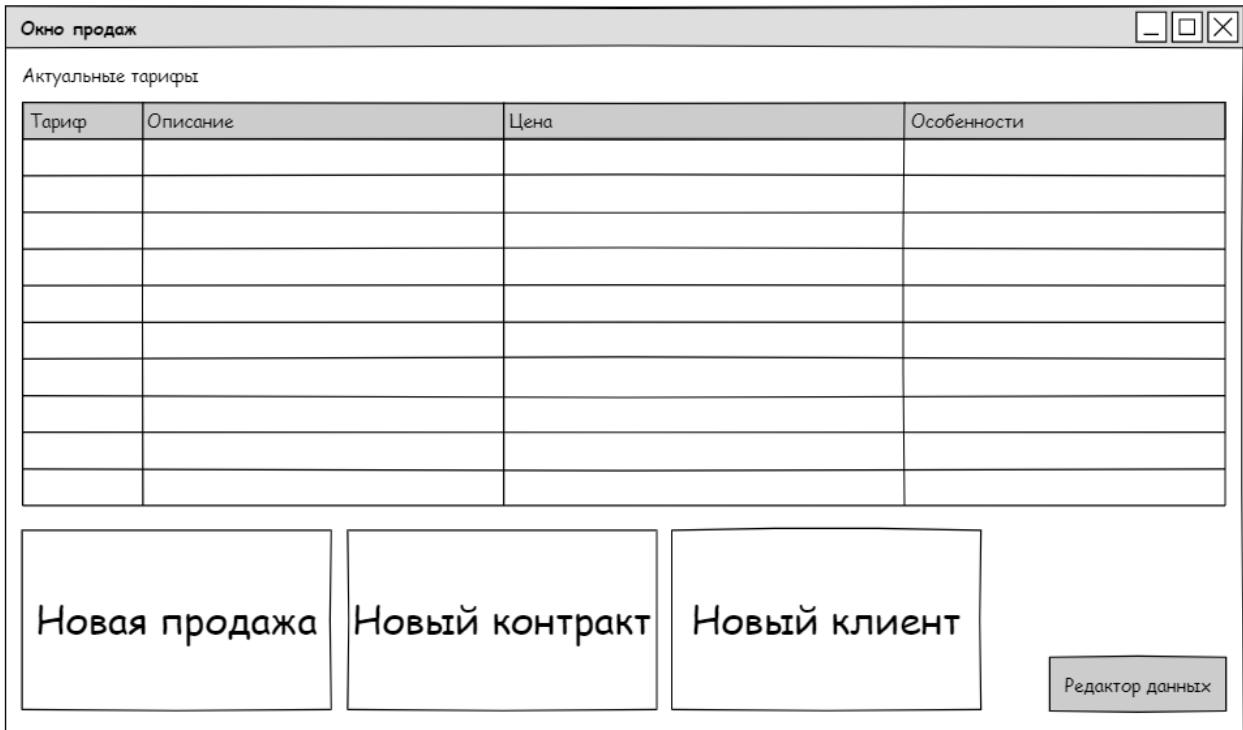


Рисунок 2.8 Макет окна продаж

На представленном макете изображено основное рабочее окно для пользователей с ролями Менеджер и Старший менеджер, а также доступное для Администратора. Это центральное окно, с которого начинается работа по оформлению сделок и управлению клиентами.

Цель окна: Предоставить пользователю быстрый доступ к ключевым функциям отдела продаж: просмотру актуальных тарифных планов, созданию новых продаж, контрактов и добавлению новых клиентов.

Интерфейс:

Окно имеет стандартную заголовочную панель с названием "Окно продаж" и кнопками управления (свернуть, развернуть, закрыть).

В верхней части окна расположен блок "Актуальные тарифы":

- Он представляет собой таблицу с четырьмя столбцами: Тариф, Описание, Цена, Особенности.

- Эта таблица отображает список всех действующих тарифных планов, доступных для подключения. Пользователь может просматривать эту информацию для консультирования клиентов.

В нижней части окна расположены три крупные кнопки-контроллера, которые являются основными точками входа в рабочие процессы:

- Новая продажа: Кнопка, при нажатии на которую открывается окно для оформления новой продажи товара клиенту.
- Новый контракт: Кнопка, при нажатии на которую открывается окно для заключения нового контракта (например, подключения к тарифу или оформления рассрочки).
- Новый клиент: Кнопка, при нажатии на которую открывается окно для регистрации нового клиента в системе.

В правом нижнем углу окна находится кнопка "Редактор данных". При нажатии на неё пользователь переходит в специализированное окно, где можно управлять справочниками системы. Доступ к этому окну возможен для всех ролей (Менеджер, Старший менеджер, Администратор), но права на редактирование будут различаться в зависимости от роли пользователя.

Этот макет обеспечивает простой и интуитивно понятный интерфейс, позволяющий пользователям быстро находить нужную информацию и начинать выполнение основных задач. Он служит стартовой площадкой для работы с клиентами и оформления сделок.

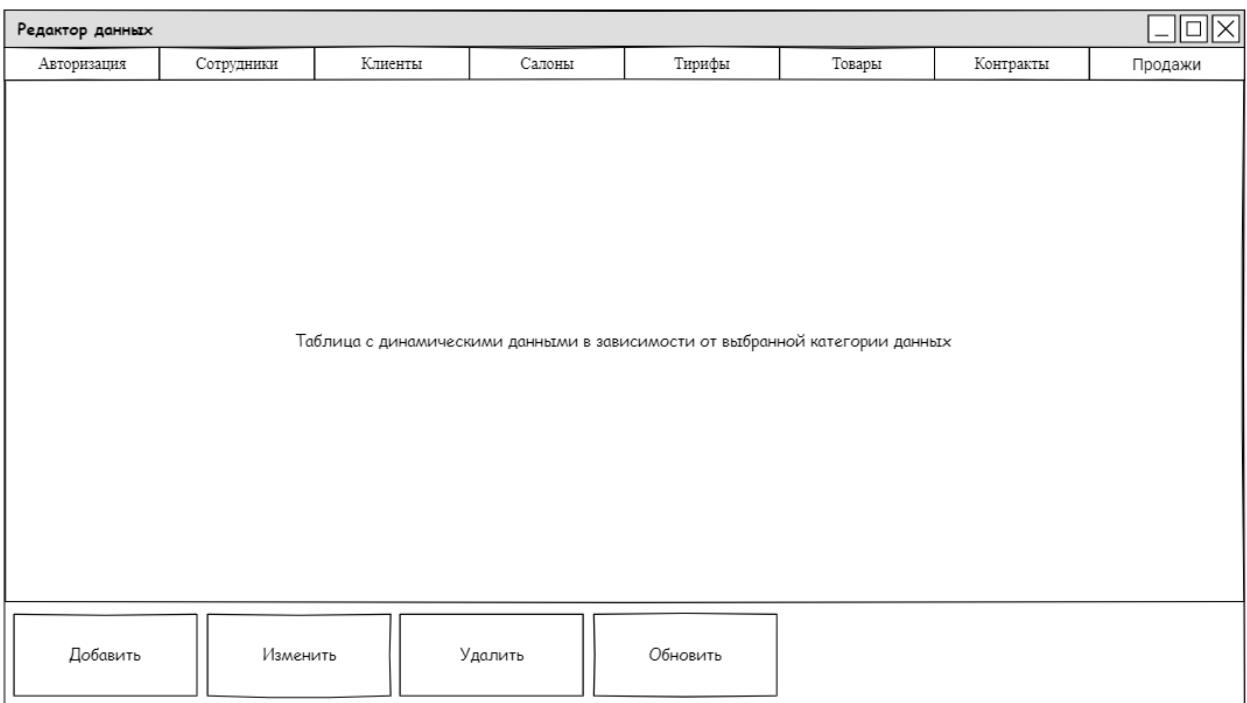


Рисунок 2.9 Макет окна «Редактор данных»

На представленном макете изображено универсальное окно "Редактор данных", которое доступно для всех пользователей системы: Менеджера, Старшего менеджера и Администратора. Это центральное место для управления всеми справочниками и данными, которые используются в системе. Окно имеет гибкую структуру, которая адаптируется под роль пользователя: не все категории данных и функции будут отображаться или быть активными для каждого актора.

Цель окна: предоставить пользователям с разными уровнями доступа возможность просматривать, добавлять, изменять и удалять данные в соответствующих справочниках, согласно их полномочиям, определенным в диаграмме Use Case.

Интерфейс:

Окно имеет стандартную заголовочную панель с названием "Редактор данных" и кнопками управления (свернуть, развернуть, закрыть).

В верхней части окна расположен набор вкладок (табов), каждая из которых представляет собой одну категорию данных:

- Авторизация
- Сотрудники

- Клиенты
- Салоны
- Тарифы
- Товары
- Контракты
- Продажи

Центральная область окна — это динамическая таблица, содержимое которой меняется в зависимости от выбранной вкладки. Эта таблица отображает список записей текущей категории (например, список всех клиентов при выборе вкладки "Клиенты").

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

- Добавить: открывает форму для создания новой записи в выбранной категории.
- Изменить: открывает форму редактирования выбранной записи (активна только если выделена строка в таблице).
- Удалить: удаляет выбранную запись (активна только если выделена строка в таблице).
- Обновить: обновляет данные в таблице, отображая последние изменения в базе данных.

Доступность для разных ролей:

Функциональность окна строго регулируется ролью пользователя, что обеспечивает безопасность и целостность данных. Кнопки Добавить, изменить, удалить и доступ к вкладкам варьируются в зависимости от роли.

Менеджер:

- Имеет доступ к вкладкам: Клиенты, Тарифы, Товары, Контракты, Продажи.
- Может выполнять операции: просмотреть.

- Может выполнить добавление в таких категориях как: Клиенты, Контракты, Продажи.

- Кнопки Изменить и удалить для Менеджера неактивны или скрыты, так как он не может изменять или удалять существующие записи.

Старший менеджер:

- Имеет доступ ко всем вкладкам, доступным Менеджеру (Клиенты, Тарифы, Товары, Контракты, Продажи).

- Также может изменить или удалить клиента, изменить данные контрактов и удалить продажи.

Администратор:

- Имеет полный доступ ко всем вкладкам: Авторизация, Сотрудники, Клиенты, Салоны, Тарифы, Товары, Контракты, Продажи.

- Может выполнять все операции: просмотреть, добавить, изменить, удалить во всех категориях (кроме изменения продажи).

- Администратор — единственный, кто может управлять самими пользователями системы (вкладка Авторизация), создавать и удалять записи о сотрудниках (вкладка Сотрудники) и салонах (вкладка Салоны), а также полностью контролировать справочники Тарифы и Товары.

Этот макет обеспечивает удобный и единый интерфейс для работы со всеми данными системы, при этом четко соблюдая принцип разграничения прав доступа. Пользователи видят только те категории и функции, которые им разрешены, что делает работу интуитивно понятной и безопасной.

Новый сотрудник	
Полное имя	<input type="text"/>
Номер телефона	<input type="text"/>
Email	<input type="text"/>
Салон	<input type="text"/> ▼
Заработка плата	<input type="text"/>
Должность	<input type="text"/>
Логин	<input type="text"/>
Пароль	*****
Повторите пароль	*****
Роль	<input type="text"/> ▼
<input type="button" value="Добавить"/>	

Рисунок 2.10 Макет окна регистрации сотрудника

На представленном макете изображено окно для создания новой записи о сотруднике в информационной системе. Это окно является частью модуля "Редактор данных" и предназначено для добавления информации о новом работнике в справочник.

Цель окна: предоставить администратору форму для ввода всех необходимых данных о новом сотруднике, которые будут сохранены в базе данных.

Интерфейс:

Окно имеет стандартную заголовочную панель с названием "Новый сотрудник" и кнопками управления (свернуть, развернуть, закрыть).

Внутри окна расположена форма, состоящая из пар: метка поля и поле ввода/выбора.

Поля формы:

- Полное имя: Текстовое поле для ввода ФИО сотрудника.
- Номер телефона: Текстовое поле для ввода контактного номера.
- Email: Текстовое поле для ввода электронной почты.
- Салон: Поле с выпадающим списком (ComboBox), позволяющее выбрать салон, к которому будет прикреплен сотрудник. Значение выбирается из заранее созданных записей в справочнике салонов.
- Заработка плата: Текстовое поле для ввода величины заработной платы.
- Должность: Текстовое поле для указания должности сотрудника.
- Логин: Текстовое поле для ввода уникального логина, который будет использоваться для входа в систему.
- Пароль: Текстовое поле для ввода пароля. Введенные символы отображаются звездочками ("*****") для обеспечения безопасности.
- Повторите пароль: Текстовое поле, в которое необходимо повторно ввести пароль для подтверждения. Это предотвращает ошибки при вводе.
- Роль: Поле с выпадающим списком (ComboBox), позволяющее назначить роль пользователю (например, "Менеджер", "Старший менеджер", "Администратор"). Значение выбирается из списка доступных ролей.

В нижней части окна расположена одна основная кнопка:

- Добавить: Кнопка, по нажатию которой система проверяет корректность введенных данных (например, совпадение паролей, заполнение обязательных полей) и сохраняет новую запись о сотруднике в базу данных. После успешного сохранения окно обычно закрывается или очищается для ввода следующего сотрудника.

Этот макет обеспечивает структурированный и удобный интерфейс для добавления новых сотрудников, что является важным элементом для управления персоналом в системе.

Форма для изменения данных сотрудника. Окно имеет заголовок "Изменить данные сотрудника" и стандартные кнопки управления (свернуть, развернуть, закрыть). Внутри расположена форма с полями для ввода информации о сотруднике:

- Полное имя
- Номер телефона
- Email
- Заработка плата
- Должность

Внизу находится кнопка "Изменить", предназначенная для сохранения изменений.

Рисунок 2.11 Макет окна редактирования данных сотрудника

На представленном макете изображено окно для редактирования существующей записи о сотруднике в информационной системе. Это окно является частью модуля "Редактор данных" и предназначено для обновления информации о работнике, уже находящемся в базе данных.

Цель окна: предоставить администратору форму для изменения данных о конкретном сотруднике, после чего обновленная информация будет сохранена в базе данных.

Интерфейс:

Окно имеет стандартную заголовочную панель с названием "Изменить данные сотрудника" и кнопками управления (свернуть, развернуть, закрыть).

Внутри окна расположена форма, состоящая из пар: метка поля и поле ввода.

Поля формы:

- Полное имя: Текстовое поле для ввода или корректировки ФИО сотрудника.
- Номер телефона: Текстовое поле для ввода или корректировки контактного номера.
- Email: Текстовое поле для ввода или корректировки электронной почты.

- Заработка плата: Текстовое поле для ввода или корректировки величины заработной платы.

- Должность: Текстовое поле для указания или изменения должности сотрудника.

В нижней части окна расположена одна основная кнопка:

- Изменить: Кнопка, по нажатию которой система проверяет корректность введенных данных и обновляет соответствующую запись в базе данных. После успешного сохранения окно обычно закрывается или очищается.

Этот макет обеспечивает структурированный и удобный интерфейс для редактирования данных сотрудников, что является важным элементом для поддержания актуальности информации в системе.

Изменить данные авторизации сотрудника	
Сотрудник	ФИО Сотрудника
Логин	<input type="text"/>
Пароль	<input type="password"/>
Роль	<input type="button" value="▼"/>
<input type="button" value="Изменить"/>	

Рисунок 2.12 Макет окна редактирования данных авторизации

На представленном макете изображено окно для редактирования учетной записи существующего сотрудника в информационной системе. Это окно является частью модуля "Редактор данных" и предназначено исключительно для администратора, так как управление авторизацией — это привилегированная функция.

Цель окна: предоставить администратору форму для изменения логина, пароля или роли пользователя (сотрудника) после того, как он был создан.

Интерфейс:

Окно имеет стандартную заголовочную панель с названием "Изменить данные авторизации сотрудника" и кнопками управления (свернуть, развернуть, закрыть).

Внутри окна расположена форма, состоящая из пар: метка поля и поле ввода/выбора.

Поля формы:

- Сотрудник: Поле, отображающее ФИО сотрудника, чьи данные будут изменены. Это поле, скорее всего, будет неизменяемым и служит для подтверждения, кого именно редактируют.
- Логин: Текстовое поле для ввода нового логина. Администратор может изменить текущий логин на другой.
- Пароль: Текстовое поле для ввода нового пароля. Как правило, для безопасности пароль вводится заново, а не показывается в открытом виде.
- Роль: Поле с выпадающим списком (ComboBox), позволяющее выбрать новую роль для сотрудника (например, с "Менеджер" на "Старший менеджер"). Значение выбирается из списка доступных ролей.

В нижней части окна расположена одна основная кнопка:

- Изменить: Кнопка, по нажатию которой система проверяет корректность введенных данных и обновляет запись о учетной записи в базе данных. После успешного сохранения окно обычно закрывается или очищается.

Этот макет обеспечивает структурированный и удобный интерфейс для администрирования пользователей системы, что является критически важным элементом для обеспечения безопасности и контроля доступа.

Форма для добавления/изменения клиента. Видимые элементы:

- Заголовок: "Добавить/Изменить клиента" в верхней левой части.
- Кнопки управления в верхней правой части: минимизация, максимизация и закрытие.
- Поля ввода:
 - "Полное имя" с текстовым полем.
 - "Номер телефона" с текстовым полем.
 - "Email" с текстовым полем.
 - "Адрес" с текстовым полем.
- Кнопка "Добавить/Изменить" в нижней части окна.

Рисунок 2.13 Макет окна добавления/изменения данных клиента

На представленном макете изображено универсальное окно для работы с записями о клиентах в информационной системе. Это окно служит как для создания новой записи, так и для редактирования существующей, что делает его ключевым инструментом для менеджеров и администраторов.

Цель окна: предоставить пользователю форму для ввода или корректировки основных данных о клиенте, которые будут сохранены в базе данных.

Интерфейс:

Окно имеет стандартную заголовочную панель с названием "Добавить/Изменить клиента" и кнопками управления (свернуть, развернуть, закрыть).

Внутри окна расположена форма, состоящая из пар: метка поля и поле ввода.

Поля формы:

- Полное имя: Текстовое поле для ввода ФИО клиента.
- Номер телефона: Текстовое поле для ввода контактного номера.
- Email: Текстовое поле для ввода электронной почты.
- Адрес: Текстовое поле для указания адреса проживания или регистрации клиента.

В нижней части окна расположена одна основная кнопка:

- Добавить/Изменить: Кнопка, по нажатию которой система проверяет корректность введенных данных и выполняет одно из двух действий:

- При создании нового клиента — добавляет новую запись в базу данных.

- При редактировании существующего клиента — обновляет данные в соответствующей записи. После успешного сохранения окно обычно закрывается или очищается для ввода следующего клиента.

Этот макет обеспечивает простой и логичный интерфейс для управления данными о клиентах, что является важным элементом для эффективного взаимодействия с клиентами и оформления сделок.

Макет окна для добавления/изменения данных о салоне. Окно имеет заголовок 'Добавить/Изменить Салон' и стандартные кнопки управления (размер, минимизация, максимизация, закрытие). Внутри окна расположены поля ввода для 'Адрес', 'Номер телефона', 'Контактное лицо' и 'Регион', каждое с собственным полем ввода. Внизу находится большая кнопка 'Добавить/Изменить'.

Рисунок 2.14 Макет окна добавления/изменения данных о салоне

На представленном макете изображено универсальное окно для работы с записями о салонах в информационной системе. Это окно служит как для создания новой записи, так и для редактирования существующей, что делает его ключевым инструментом для администратора.

Цель окна: предоставить администратору форму для ввода или корректировки основных данных о салоне, которые будут сохранены в базе данных.

Интерфейс:

Окно имеет стандартную заголовочную панель с названием "Добавить/Изменить Салон" и кнопками управления (свернуть, развернуть, закрыть).

Внутри окна расположена форма, состоящая из пар: метка поля и поле ввода.

Поля формы:

- Адрес: Текстовое поле для ввода полного адреса салона.
- Номер телефона: Текстовое поле для ввода контактного номера салона.
- Контактное лицо: Текстовое поле для указания ФИО ответственного лица в салоне.
- Регион: Текстовое поле для ввода наименования региона, в котором находится салон.

В нижней части окна расположена одна основная кнопка:

- Добавить/Изменить: Кнопка, по нажатию которой система проверяет корректность введенных данных и выполняет одно из двух действий:
 - При создании нового салона — добавляет новую запись в базу данных.
 - При редактировании существующего салона — обновляет данные в соответствующей записи. После успешного сохранения окно обычно закрывается или очищается для ввода следующего салона.

Этот макет обеспечивает простой и логичный интерфейс для управления данными о салонах, что является важным элементом для эффективного администрирования сети салонов в системе.

Форма для добавления/изменения тарифа, состоящая из следующих полей:

- Название**: Текстовое поле для ввода наименования тарифного плана.
- Описание**: Текстовое поле для ввода подробного описания условий тарифа (включенные услуги, лимиты, особенности).
- Цена**: Текстовое поле для ввода стоимости тарифного плана.
- Особенности**: Текстовое поле для ввода специальных характеристик тарифа.

Внизу расположена кнопка **Добавить/Изменить**.

Рисунок 2.15 Макет окна добавления/изменения данных тарифов

На представленном макете изображено универсальное окно для работы с записями о тарифных планах в информационной системе. Это окно служит как для создания новой записи, так и для редактирования существующей, что делает его ключевым инструментом для администратора и старшего менеджера.

Цель окна: предоставить пользователю форму для ввода или корректировки основных данных о тарифном плане, которые будут сохранены в базе данных.

Интерфейс:

Окно имеет стандартную заголовочную панель с названием "Добавить/Изменить Тариф" и кнопками управления (свернуть, развернуть, закрыть).

Внутри окна расположена форма, состоящая из пар: метка поля и поле ввода.

Поля формы:

- **Название**: Текстовое поле для ввода краткого наименования тарифного плана.
- **Описание**: Текстовое поле для ввода подробного описания условий тарифа (включенные услуги, лимиты, особенности).
- **Цена**: Текстовое поле для ввода стоимости тарифного плана.

- Особенности: Текстовое поле для указания уникальных особенностей тарифа (например, безлимитные звонки, бонусы, акции).

В нижней части окна расположена одна основная кнопка:

- Добавить/Изменить: Кнопка, по нажатию которой система проверяет корректность введенных данных и выполняет одно из двух действий:

○ При создании нового тарифа — добавляет новую запись в базу данных.

○ При редактировании существующего тарифа — обновляет данные в соответствующей записи. После успешного сохранения окно обычно закрывается или очищается для ввода следующего тарифа.

Этот макет обеспечивает простой и логичный интерфейс для управления справочником тарифных планов, что является важным элементом для эффективного консультирования клиентов и оформления контрактов.

Макет окна для добавления/изменения товаров. Окно имеет заголовок 'Добавить/Изменить Товар' и стандартные кнопки управления (размер, минимизация, закрытие). Внутри окна расположены шесть полей для ввода информации о товаре: 'Название', 'Тип', 'Производитель', 'Модель', 'Цена' и 'Гарантия'. Каждое поле имеет соответствующую метку слева. Внизу окна находится одна большая кнопка 'Добавить/Изменить'.

Рисунок 2.16 Макет окна добавления/изменения товаров

На представленном макете изображено универсальное окно для работы с записями о товарах в информационной системе. Это окно служит как для создания новой записи, так и для редактирования существующей, что делает его ключевым инструментом для администратора и старшего менеджера.

Цель окна: предоставить пользователю форму для ввода или корректировки основных данных о товаре, которые будут сохранены в базе данных.

Интерфейс:

Окно имеет стандартную заголовочную панель с названием "Добавить/Изменить Товар" и кнопками управления (свернуть, развернуть, закрыть).

Внутри окна расположена форма, состоящая из пар: метка поля и поле ввода.

Поля формы:

- Название: Текстовое поле для ввода общего наименования товара.
- Тип: Текстовое поле для указания категории товара (например, "Смартфон", "Аксессуар", "SIM-карта").
- Производитель: Текстовое поле для ввода названия компании-производителя.
- Модель: Текстовое поле для ввода конкретной модели товара.
- Цена: Текстовое поле для указания стоимости товара.
- Гарантия: Текстовое поле для указания срока гарантии (например, "12 месяцев").

В нижней части окна расположена одна основная кнопка:

- Добавить/Изменить: Кнопка, по нажатию которой система проверяет корректность введенных данных и выполняет одно из двух действий:
 - При создании нового товара — добавляет новую запись в базу данных.
 - При редактировании существующего товара — обновляет данные в соответствующей записи. После успешного сохранения окно обычно закрывается или очищается для ввода следующего товара.

Этот макет обеспечивает простой и логичный интерфейс для управления справочником товаров, что является важным элементом для эффективного консультирования клиентов и оформления продаж.

Форма окна 'Добавить/Изменить Контракт' имеет следующую структуру:

- Заголовок:** 'Добавить/Изменить Контракт' в верхней левой части.
- Кнопки управления:** в верхней правой части находятся кнопки для минимизации, максимизации и закрытия окна.
- Поля ввода:**
 - 'Клиент': Поле с выпадающим списком (ComboBox).
 - 'Тариф': Поле с выпадающим списком (ComboBox).
 - 'Статус': Текстовое поле.
- Кнопка:** 'Добавить/Изменить' расположена в нижней части окна.

Рисунок 2.17 Макет окна добавить/изменить контракт

На представленном макете изображено универсальное окно для работы с записями о контрактах в информационной системе. Это окно служит как для создания новой записи, так и для редактирования существующей, что делает его ключевым инструментом для менеджеров и администраторов.

Цель окна: предоставить пользователю форму для ввода или корректировки основных данных о контракте, которые будут сохранены в базе данных.

Интерфейс:

Окно имеет стандартную заголовочную панель с названием "Добавить/Изменить Контракт" и кнопками управления (свернуть, развернуть, закрыть).

Внутри окна расположена форма, состоящая из пар: метка поля и поле ввода/выбора.

Поля формы:

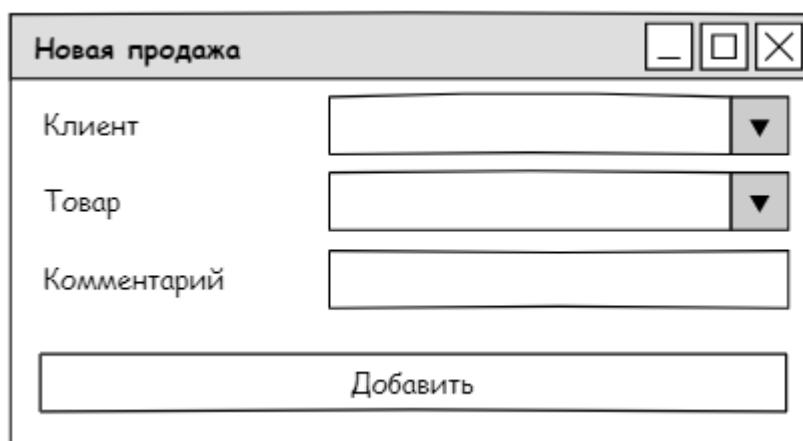
- Клиент: Поле с выпадающим списком (ComboBox), позволяющее выбрать клиента из справочника. Значение выбирается из заранее созданных записей.
- Тариф: Поле с выпадающим списком (ComboBox), позволяющее выбрать тарифный план из справочника. Значение выбирается из заранее созданных записей.
- Статус: Текстовое поле или поле с выпадающим списком для указания текущего статуса контракта (например, "Активен", "Завершен", "Расторгнут").

В нижней части окна расположена одна основная кнопка:

- Добавить/Изменить: Кнопка, по нажатию которой система проверяет корректность введенных данных и выполняет одно из двух действий:

- При создании нового контракта — добавляет новую запись в базу данных.
- При редактировании существующего контракта — обновляет данные в соответствующей записи. После успешного сохранения окно обычно закрывается или очищается для ввода следующего контракта.

Этот макет обеспечивает простой и логичный интерфейс для оформления и управления контрактами, что является важным элементом для эффективного взаимодействия с клиентами и учета услуг.



Макет окна добавления продажи (Новая продажа) показывает следующее содержание:

- Заголовок: Новая продажа
- Контрольные кнопки: свернуть, развернуть, закрыть
- Поля ввода:
 - Клиент: выпадающий список
 - Товар: выпадающий список
 - Комментарий: поле ввода
- Кнопка: Добавить

Рисунок 2.18 Макет окна добавления продажи

На представленном макете изображено окно для оформления новой продажи товара клиенту. Это ключевое рабочее окно, с которым взаимодействуют менеджеры и администраторы при совершении сделки.

Цель окна: предоставить пользователю форму для быстрого и удобного оформления записи о продаже, включающей выбор клиента, товара и добавление комментариев.

Интерфейс:

Окно имеет стандартную заголовочную панель с названием "Новая продажа" и кнопками управления (свернуть, развернуть, закрыть).

Внутри окна расположена форма, состоящая из пар: метка поля и поле ввода/выбора.

Поля формы:

- Клиент: Поле с выпадающим списком (ComboBox), позволяющее выбрать клиента из справочника. Значение выбирается из заранее созданных записей.
- Товар: Поле с выпадающим списком (ComboBox), позволяющее выбрать товар из справочника. Значение выбирается из заранее созданных записей.
- Комментарий: Текстовое поле для ввода дополнительной информации по продаже (например, условия оплаты, особые пожелания клиента).

В нижней части окна расположена одна основная кнопка:

- Добавить: Кнопка, по нажатию которой система проверяет корректность введенных данных и сохраняет новую запись о продаже в базу данных. После успешного сохранения окно обычно закрывается или очищается для оформления следующей продажи.

Этот макет обеспечивает простой и логичный интерфейс для быстрого оформления продаж, что является важным элементом для эффективной работы отдела продаж.

2.5. Описание логики функционирования ПО

Планируемая ИС представляет собой десктопное приложение, предназначенное для автоматизации ключевых бизнес-процессов отдела продаж. Логика функционирования ПО будет направлена на обеспечение эффективного взаимодействия между пользователями системы, обладающими различными ролями (Администратор, Старший менеджер, Менеджер), и централизованной базой данных, содержащей сведения о клиентах, сотрудниках, товарах, тарифных планах, продажах и контрактах. Система будет спроектирована с учётом требований к безопасности, производительности, удобству использования и масштабируемости.

Планируемая архитектура ПО будет соответствовать клиент-серверной модели, где десктопное клиентское приложение будет взаимодействовать с

сервером базы данных. Архитектура будет спроектирована по слоистому принципу (multi-tier), что обеспечит модульность, облегчит сопровождение и масштабирование в будущем.

1) Пользовательский интерфейс (UI Layer): будет реализован с использованием выбранной технологии (например, Windows Forms в .NET). Включит в себя окна, описанные в макетах (см. п. 2.4), такие как "Авторизация", "Окно продаж", "Редактор данных" и другие. Интерфейс будет спроектирован с учётом требований к удобству использования, обеспечивая интуитивно понятную навигацию и логичную структуру элементов управления.

2) Слой приложения (Application/Business Logic Layer): В этом слое будет реализована основная логика работы системы. Он будет отвечать за обработку команд пользователя, вызов методов для работы с данными, выполнение бизнес-правил (например, проверка корректности введенных данных, соблюдение логики бизнес-процессов, разграничение доступа в зависимости от роли пользователя) и формирование ответа для отображения в UI. Этот слой будет включать классы, управляющие сессией пользователя, валидацией данных и т.д.

3) Слой доступа к данным (Data Access Layer): Этот компонент будет отвечать за взаимодействие с сервером базы данных (например, Microsoft SQL Server). Он будет реализовывать операции CRUD (Create, Read, Update, Delete) для взаимодействия с таблицами базы данных, обеспечивая фильтрацию, сортировку и передачу данных между базой и слоем приложения.

Такая архитектура позволит чётко разделить ответственность между компонентами, упростит тестирование и модификацию отдельных частей системы на этапе разработки и в дальнейшем.

2.5.1. Вход в систему и разграничение прав

Процесс взаимодействия с системой начнётся с аутентификации пользователя. При запуске приложения пользователь будет видеть окно

"Авторизация". Здесь он должен будет ввести логин и пароль. Система будет проверять введенные данные в базе данных.

- При успешной проверке система определит роль пользователя (например, "Менеджер", "Старший менеджер", "Администратор"), привязанную к его учетной записи.
- На основе определенной роли система загрузит главное окно, соответствующее этой роли, и динамически скроет или сделает неактивными элементы интерфейса, недоступные для данной роли. Например, кнопки "Удалить" или "Редактировать" в "Редакторе данных" будут активны для Старшего менеджера и Администратора, но неактивны для Менеджера. Окно "Администрация" будет доступно только Администратору. Это обеспечит разграничение прав доступа в соответствии с диаграммой Use Case.

2.5.2. Работа с клиентами

Пользователь сможет взаимодействовать с данными о клиентах через главное окно "Окно продаж" (для Менеджера, Старшего менеджера) или через "Редактор данных".

- Добавление клиента: при нажатии кнопки "Новый клиент" в "Окне продаж" откроется форма "Добавить/Изменить клиента". Пользователь будет вводить данные (ФИО, телефон, email, адрес) и нажимать "Добавить". Система передаст данные в слой приложения, который их валидирует и, при успешной проверке, через слой доступа к данным, добавит новую запись в таблицу Client базы данных.
- Просмотр/редактирование: В "Редакторе данных", при выборе вкладки "Клиенты", система загрузит все записи из таблицы БД и отобразит их в таблице окна. Менеджер сможет просматривать данные. Старший менеджер и Администратор смогут выделить строку с клиентом и нажать "Изменить", что откроет форму "Добавить/Изменить клиента" с уже заполненными полями. После внесения изменений и нажатия кнопки "Изменить" система обновит соответствующую запись в базе данных. Администратор сможет также удалить запись.

2.5.3. Работа с продажами и контрактами

- Оформление продажи: В "Окне продаж" пользователь сможет нажать кнопку "Новая продажа". Откроется форма "Новая продажа". Пользователь выберет клиента и товар из выпадающих списков и сможет добавить комментарий. При нажатии "Добавить" система создаст новую запись в БД, фиксируя ID клиента, ID товара, дату продажи, сумму и комментарий.
- Оформление контракта: аналогично, через "Окно продаж" или "Редактор данных" > вкладка "Контракты", пользователь сможет создать контракт. Откроется форма "Добавить/Изменить Контракт". Пользователь выберет клиента и тарифный план из списков и укажет статус. При сохранении будет создана новая запись в таблице.

2.5.4. Работа со справочниками и аналитика

Пользователи с соответствующими правами (в основном через "Редактор данных") смогут управлять справочниками: Товары, Тарифы, Сотрудники, Салоны. Логика будет схожа с работой с клиентами: выбор вкладки > отображение таблицы > использование кнопок "Добавить", "Изменить", "Удалить" (доступность зависит от роли). Данные будут передаваться через слой приложения в слой доступа к данным для выполнения соответствующих операций в соответствующих таблицах БД.

В окне "Администрация" (доступно только Администратору) будет реализована функция аналитики. При выборе салона из списка и нажатии кнопки "Анализ", система в слое приложения сформирует SQL-запрос к базе данных, группируя и суммируя данные о продажах (Sale) за текущий и предыдущий месяцы для выбранного салона (через связь с Employee и ID_Salon). Результаты (например, общая выручка) будут отображаться в основной области окна, позволяя Администратору сравнивать эффективность работы салонов.

Описанная логика функционирования определяет, как будет работать разрабатываемое ПО после его создания. Она чётко структурирована, основана на требованиях и диаграммах, и предусматривает разграничение

прав доступа в соответствии с ролями пользователей. Такая логика позволит в будущем создать эффективную, надёжную и удобную в сопровождении систему, способную автоматизировать ключевые аспекты деятельности отдела продаж салона сотовой связи.

3. СПЕЦИАЛЬНАЯ ЧАСТЬ

3.1. Формирование требований к ПО

Формирование требований к программному обеспечению (ПО) является критически важным этапом в процессе разработки информационной системы отдела продаж салонов сотовой связи. Этот этап 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.

Целью формирования требований в рамках данной выпускной квалификационной работы является создание чёткого и однозначного технического задания, которое служит основой для всех участников процесса разработки: разработчиков, тестировщиков и, в конечном итоге, заказчика.

Для разработчиков это определяет, какие функции необходимо реализовать и какими характеристиками должна обладать система. Для тестировщиков — это критерии оценки соответствия готового продукта спецификации. Для заинтересованных сторон — документ, подтверждающий, что создаваемое ПО решает поставленные задачи.

От качества и полноты сформулированных требований зависит успех всего проекта: архитектура системы, функциональность, удобство использования, стоимость разработки и сроки 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.

Процесс формирования требований опирается на результаты предыдущих этапов проекта, в первую очередь, на анализ предметной области, где были выявлены проблемы текущих бизнес-процессов и обозначена необходимость автоматизации. Также на него повлияли результаты сравнительного анализа систем-аналогов, подтвердившего целесообразность разработки индивидуального решения, и разработанные диаграммы,

уточнившие логику взаимодействия и структуру данных. 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 отдела продаж салонов сотовой связи к функциональным требованиям относятся:

- Управление клиентами: Возможность добавления, просмотра, редактирования и поиска информации о клиентах (ФИО, контактные данные, история покупок/подключений).
- Управление сотрудниками: Возможность добавления, просмотра, редактирования и поиска информации о сотрудниках (ФИО, должность, салон, контактные данные). (Доступность операций зависит от роли пользователя).
- Управление товарами: Возможность добавления, просмотра, редактирования и поиска информации о товарах (название, тип, производитель, модель, цена, гарантия). (Доступность операций зависит от роли пользователя).
- Управление тарифными планами: Возможность добавления, просмотра, редактирования и поиска информации о тарифных планах и услугах (название, описание, стоимость, особенности). (Доступность операций зависит от роли пользователя).
- Управление продажами: Возможность оформления новых продаж (выбор клиента, товара), просмотра истории продаж, фильтрации и поиска. (Доступность операций (например, редактирование, удаление) зависит от роли пользователя).
- Управление контрактами: Возможность оформления новых контрактов (выбор клиента, тарифа), просмотра истории контрактов, фильтрации и поиска, изменения статуса. (Доступность операций (например, редактирование, удаление) зависит от роли пользователя).

- Система авторизации и ролевой модели: Обязательная аутентификация пользователя (ввод логина и пароля) для доступа к системе. Реализация ролевой модели с разными уровнями доступа (например, "Менеджер", "Старший менеджер", "Администратор"), определяющей, какие функции и данные доступны конкретному пользователю.
- Аналитика: Возможность для администратора проводить сравнительный анализ продаж по салонам за разные периоды времени.

Нефункциональные требования 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 типичные запросы (просмотр списка клиентов, товаров, продаж) в течение 1-2 секунд при одновременной работе до 10 пользователей.
- Надёжность: Система должна быть устойчива к сбоям и обеспечивать целостность данных при корректной эксплуатации. Обязательно наличие механизма резервного копирования базы данных.
- Безопасность: Пароли пользователей должны храниться в зашифрованном виде. Реализация разграничения доступа к данным и функциям в зависимости от роли пользователя. Желательно ведение логов действий пользователей.
- Удобство использования (Usability): Интерфейс должен быть интуитивно понятным, соответствовать общепринятым стандартам. Интерфейс должен быть локализован на русский язык. Система должна предоставлять пользователю понятные сообщения об ошибках или успешных действиях.
- Совместимость: Система должна корректно работать в предполагаемой операционной системе (например, Windows 10/11).

- Масштабируемость: Архитектура системы должна предусматривать возможность добавления новых функций и увеличения числа пользователей в будущем.

Таким образом, процесс формирования требований к ПО для информационной системы отдела продаж салонов сотовой связи 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 одним из ключевых этапов в создании информационной системы, определяющим её удобство, эффективность и общую воспринимаемую ценность для конечного пользователя. Интерфейс — это "лицо" системы, через которое пользователь взаимодействует с логикой приложения и данными. От его качества напрямую зависит скорость освоения программы, точность выполнения операций и удовлетворенность работой с системой. Непродуманный или неудобный интерфейс может свести на нет все усилия по разработке мощной функциональности.

В рамках данной выпускной квалификационной работы процесс разработки интерфейса основывался на ранее сформулированных требованиях к системе (см. п. 2.1) и макетах, разработанных на этапе проектирования (см. п. 2.4). Макеты служили визуальным ориентиром, определяя структуру окон, расположение элементов управления и логику навигации. Целью разработки было создание интуитивно понятного, логичного и функционального интерфейса, который соответствовал бы потребностям пользователей с различными ролями: Менеджера, Старшего менеджера и Администратора.

Особое внимание уделялось обеспечению разграничения прав доступа, чтобы каждый пользователь видел только те функции и данные, которые ему доступны.

Ниже приведены реальные снимки экрана реализованного программного обеспечения, демонстрирующие окна и формы, которые были созданы в соответствии с проектными макетами. Эти снимки позволяют оценить, как теоретические решения и дизайнерские идеи были воплощены в практике, и показывают конечный результат работы над пользовательским интерфейсом.

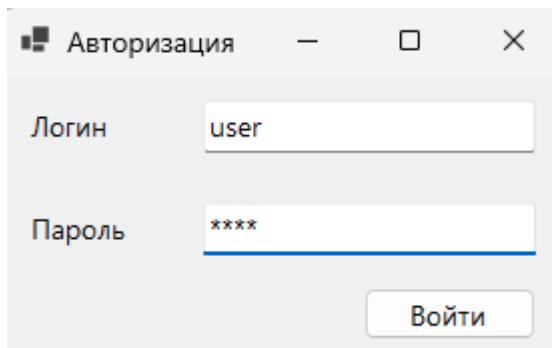


Рисунок 3.1 Окно авторизации

Администрация

Аналитика выберите салон для анализа ул. Ленина, д. 11, Москва

Анализ

Аналитика для салона на: ул. Ленина, д. 11, Москва

Периоды: октябрь 2025 vs ноябрь 2025

Анализ клиентов

Новые клиенты в текущем месяце: 8

Новые клиенты в предыдущем месяце: 11

Изменение: -3 (Снижение)

Анализ продаж

Продаж в текущем месяце: 14

Продаж в предыдущем месяце: 11

Изменение количества продаж: +3 (Рост)

Общая сумма продаж в текущем месяце: 1 260 500,00 ₽

Панель продаж >>> Редактор данных >>>

A screenshot of a web-based administrative interface. The title bar says "Администрация". Below it, a dropdown menu shows "Аналитика" selected, with "ул. Ленина, д. 11, Москва" chosen. A blue "Анализ" button is on the right. The main content area displays analysis results for a salon located at ul. Leninga, d. 11, Moscow, comparing October 2025 and November 2025. It shows statistics for new clients and sales volume, with a note about a decrease in new clients and an increase in sales.

Рисунок 3.2 Окно администратора

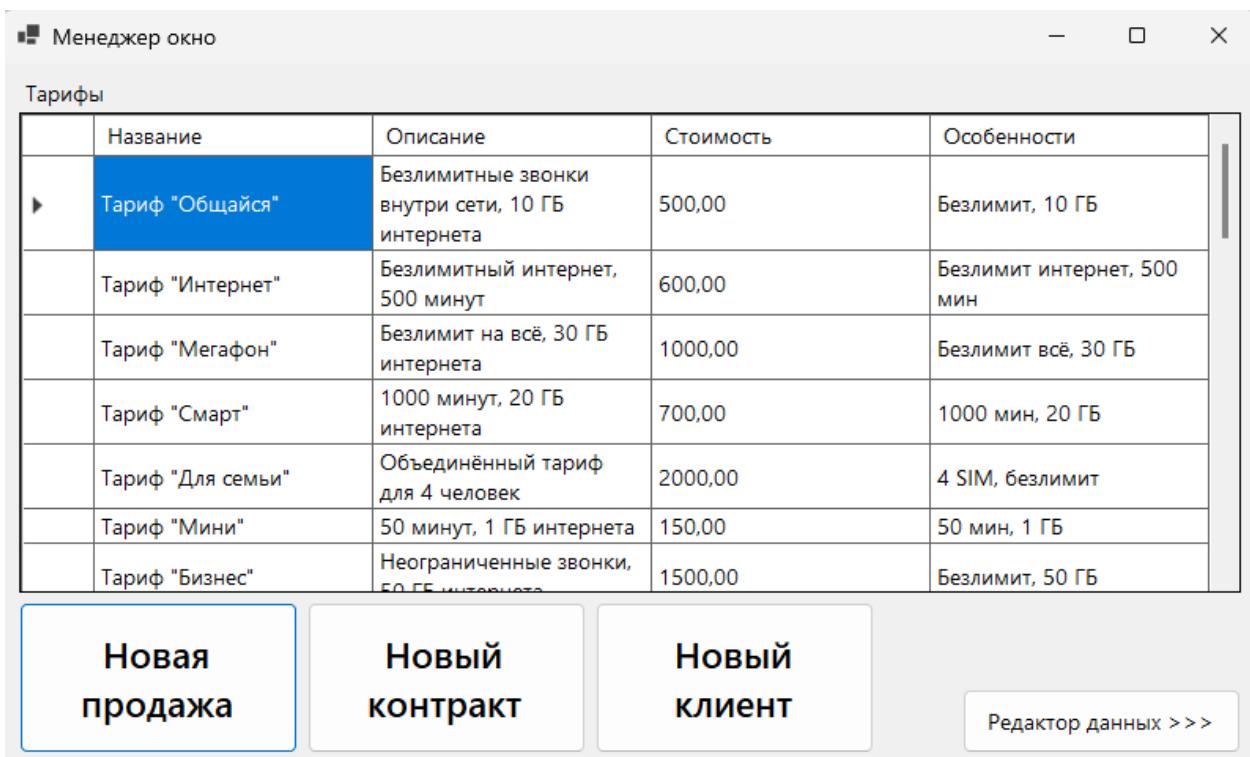


Рисунок 3.3 Окно продаж «Окно менеджера»

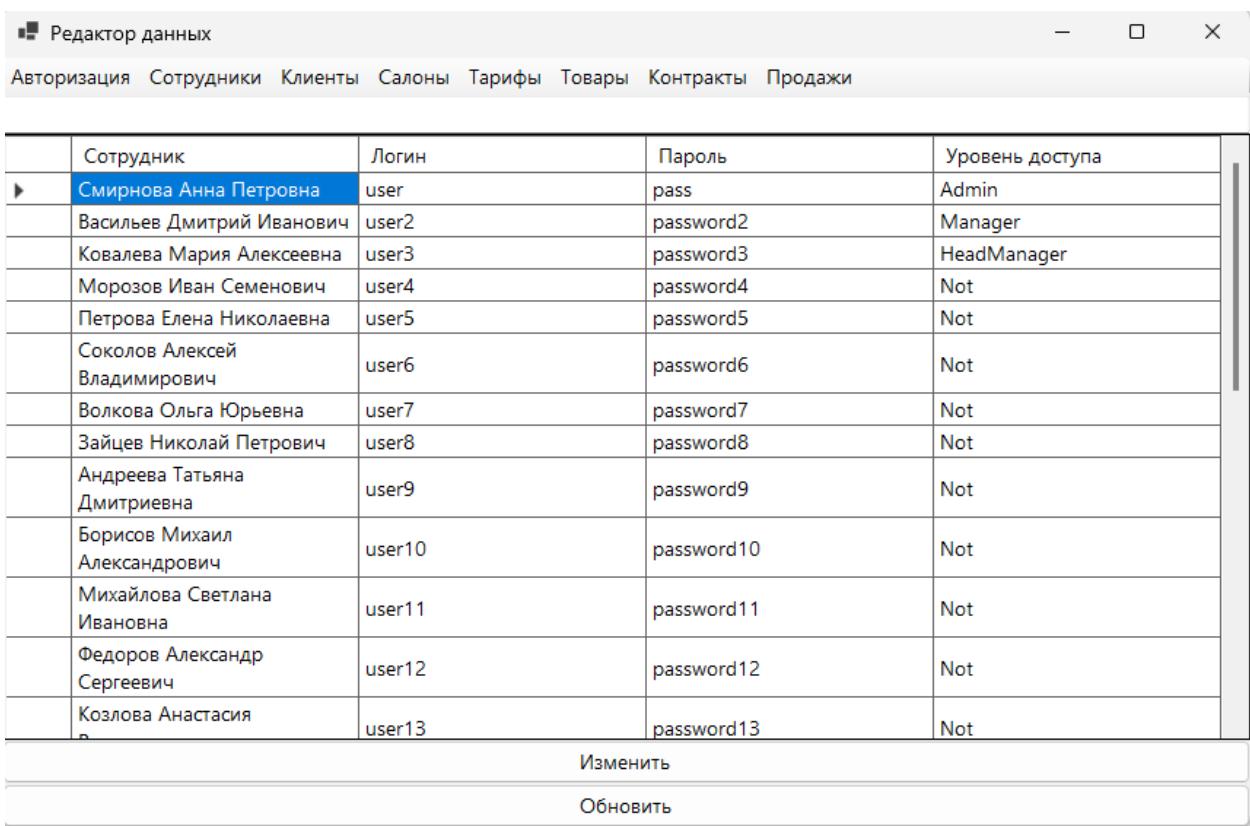


Рисунок 3.4 Окно «Редактор данных (администратор)»

Авторизация - Изменить

Сотрудник	Смирнова Анна Петровна
Логин	user
Пароль	pass
Уровень доступа	<input type="text" value="Admin"/> Admin Admin HeadManager Manager Not

Изменить

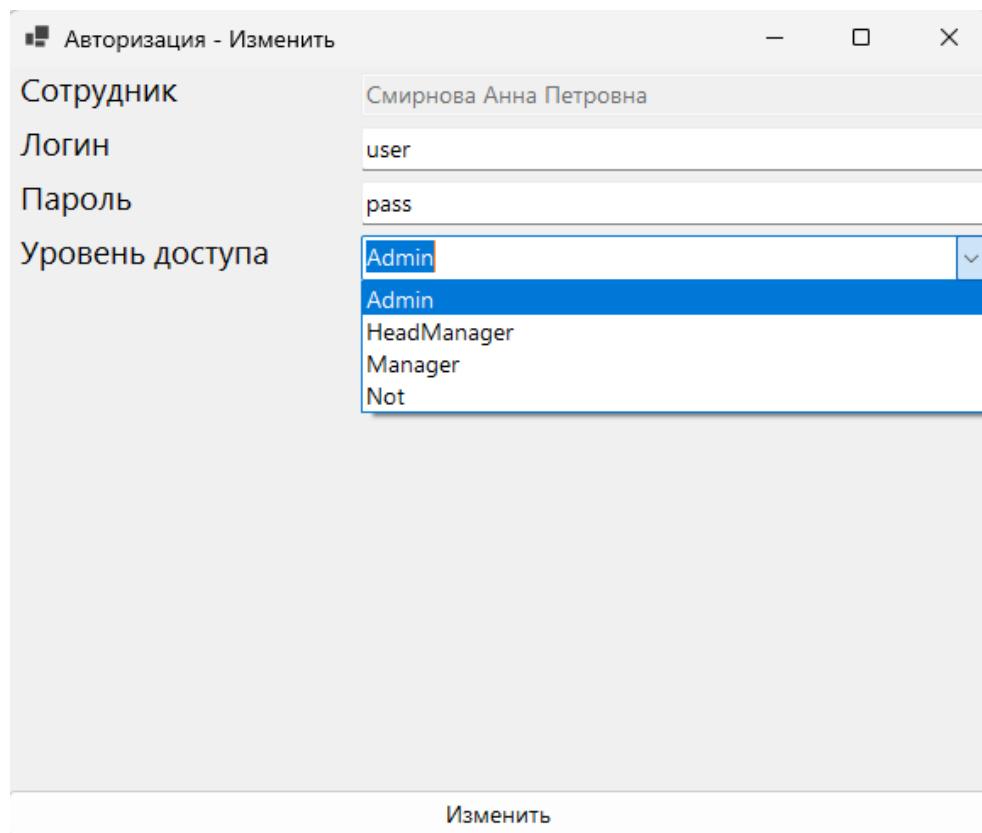


Рисунок 3.5 Окно изменения данных авторизации

Сотрудники - Изменить

ФИО	Смирнова Анна Петровна
Должность	Продавец-консультант
Номер телефона	+79151234567
Электронная почта	smirnova@example.com
Заработка плата	50000,00
Салон	ул. Ленина, д. 11, Москва

Изменить

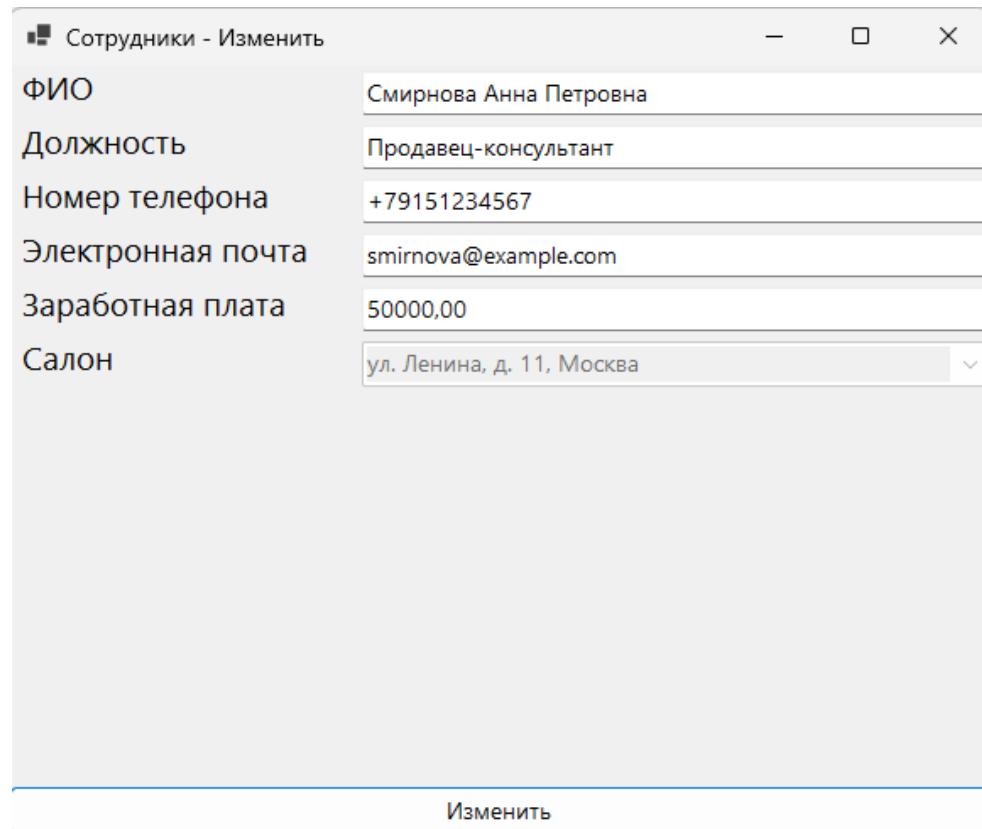


Рисунок 3.6 Окно изменения данных сотрудника

■ Регистрация сотрудника

Для регистрации введите следующие данные

Полное имя:	
Номер телефона:	
Email:	
Салон:	пр. Ленина, д. 30, Казань
Заработка плата:	0,00
Должность:	
Логин:	
Пароль:	
Повторите пароль:	
Роль:	
Зарегистрировать	

Рисунок 3.7 Окно регистрации нового сотрудника

■ Клиента - Изменить

ФИО	Кузнецов Александр Сергеевич
Номер телефона	+79251234567
Электронная почта	kuznetsov@example.com
Адрес	ул. Пушкина, д. 1, Москва

Изменить

Рисунок 3.8 Окно изменения данных клиента

■ Салоны - Изменить

Адрес	ул. Ленина, д. 11, Москва
Номер телефона	+74951234567
Контактное лицо	Иванов И.И.
Регион	Москва

Изменить

Рисунок 3.9 Окно изменения данных салона

■ Тарифы - Изменить

Наименование	Тариф "Общайся"
Описание	Безлимитные звонки внутри сети, 10 ГБ интернета
Цена	500,00
Особенности	Безлимит, 10 ГБ

Изменить

Рисунок 3.10 Окно изменения данных тарифа

■ Товары - Изменить

Наименование	iPhone 15 Pro
Тип	Телефон
Производитель	Apple
Модель	iPhone 15 Pro
Цена	110000,00
Гарантия	12 месяцев

[Изменить](#)

Рисунок 3.11 Окно изменения данных о товаре

■ Контракты - Изменить

Клиент	Кузнецов Александр Сергеевич
Тариф	Тариф "Общайся"
Статус	Активен

[Изменить](#)

Рисунок 3.12 Окно изменения данных контракта

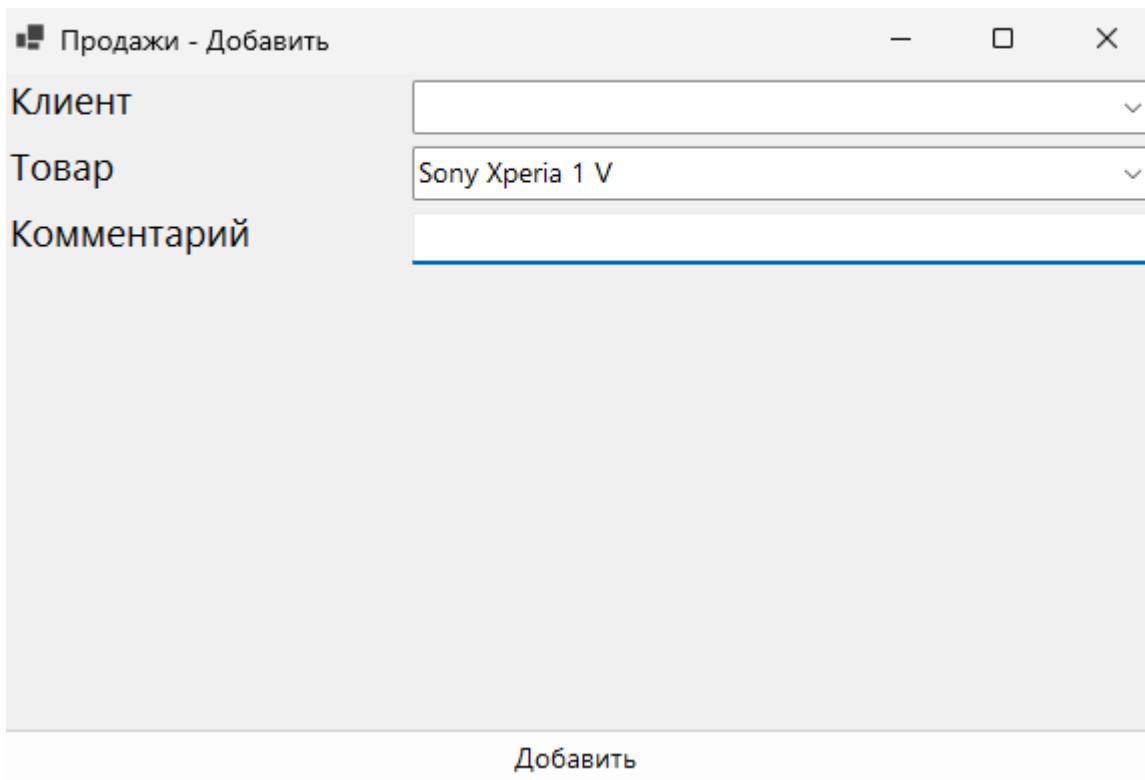


Рисунок 3.13 Окно добавления новой продажи

Редактор данных					
	ФИО	Номер телефона	Электронная почта	Адрес	Дата регистрации
▶	Кузнецов Александр Сергеевич	+79251234567	kuznetsov@example.com	ул. Пушкина, д. 1, Москва	10.01.2022 0:00:00
	Лебедева Ольга Петровна	+79251234568	lebedeva@example.com	пр. Мира, д. 2, Санкт-Петербург	15.02.2022 0:00:00
	Новиков Дмитрий Иванович	+79251234569	novikov@example.com	ул. Гагарина, д. 3, Новосибирск	20.03.2022 0:00:00
	Морозова Мария Алексеевна	+79251234570	morozova@example.com	ул. Кирова, д. 4, Екатеринбург	25.04.2022 0:00:00
	Федоров Иван Семенович	+79251234571	fedorov@example.com	пр. Ленина, д. 5, Казань	30.05.2022 0:00:00
	Волкова Елена Николаевна	+79251234572	volkova@example.com	ул. Советская, д. 6, Нижний Новгород	05.06.2022 0:00:00
	Алексеев Алексей Владимирович	+79251234573	alekseev@example.com	ул. Пушкина, д. 7, Краснодар	10.07.2022 0:00:00
	Орлова Татьяна Дмитриевна	+79251234574	orlova@example.com	ул. Лермонтова, д. 8, Воронеж	15.08.2022 0:00:00
	Комаров Михаил Александрович	+79251234575	komarov@example.com	ул. Чехова, д. 9, Ростов-на-Дону	20.09.2022 0:00:00
	Киселева Светлана	+79251234576		ул. Победы, д. 10,	25.10.2022 0:00:00

Рисунок 3.14 Окно «Редактор данных (менеджер)»

3.3. Проектирование структуры базы данных

База данных (БД) представляет собой организованное и структурированное хранилище информации, обычно представленное в

электронном виде на компьютере или сервере. Основное её назначение — обеспечить удобный, надёжный и эффективный способ хранения, поиска, управления и обработки больших объёмов данных. В отличие от разрозненных файлов или бумажных носителей, база данных объединяет информацию в единую логически связанную систему, что значительно упрощает работу с ней и минимизирует риски потерь или дублирования.

Для информационной системы отдела продаж салонов сотовой связи, база данных выступает в роли центрального ядра, где хранятся все ключевые сведения: данные о клиентах, сотрудниках, товарах, тарифных планах, продажах и контрактах. Правильно спроектированная и реализованная БД является фундаментом для стабильной и корректной работы всего программного обеспечения.

Основные цели базы данных:

1) Эффективное хранение информации: обеспечить структурированное, логичное и надёжное хранение больших объёмов данных, характерных для деятельности салона сотовой связи.

2) Управление данными: предоставить программному обеспечению и пользователям средства для выполнения основных операций с данными: добавления новых записей (INSERT), чтения (SELECT), изменения (UPDATE) и удаления (DELETE).

3) Обеспечение доступа: обеспечить быстрый, контролируемый и последовательный доступ к информации как для пользователей системы (с учётом их ролей), так и для прикладных программ.

4) Поддержание целостности и безопасности: обеспечить точность, согласованность и достоверность данных, а также защитить их от несанкционированного доступа, изменений или потерь.

5) Снижение избыточности: минимизировать дублирование информации за счёт нормализации структуры, когда каждая сущность хранится в одном месте, а связи между ними реализуются через специальные ключи.

Основные элементы базы данных:

1) Таблицы (Tables): Фундаментальный компонент реляционной базы данных. Таблица структурно напоминает электронную таблицу, состоящую из строк и столбцов. Каждая таблица предназначена для хранения информации об одном типе сущности. В рамках проекта это могут быть такие таблицы, как Client (для хранения данных о клиентах), Employee (для данных о сотрудниках), Product (для товаров), TariffPlan (для тарифов), Sale (для записей о продажах) и Contract (для контрактов).

2) Поля (Fields / Columns): Столбцы таблицы. Каждое поле определяет конкретный атрибут или характеристику сущности, описываемой таблицей. Например, в таблице Client полями могут быть ID_Client, FullName, Phone, Email, Address.

3) Записи (Records / Rows): Строки таблицы. Каждая запись представляет собой один конкретный экземпляр сущности и содержит набор значений для всех полей таблицы. Например, одна строка в таблице Employee будет содержать конкретную информацию об одном сотруднике (его ID, ФИО, должность, телефон и т.д.).

4) Ключи (Keys):

- Первичный ключ (Primary Key): Уникальный идентификатор каждой строки в таблице. Обычно это числовое поле с автоматическим увеличением (IDENTITY), которое гарантирует, что ни одна запись не будет дублировать другую по значению первичного ключа. Примеры: ID_Client в таблице Client, ID_Sale в таблице Sale.

- Внешний ключ (Foreign Key): Поле (или набор полей), значение которого соответствует первичному ключу в другой (или той же) таблице. Внешние ключи устанавливают логические связи между таблицами, обеспечивая целостность данных. Например, поле ID_Client в таблице Sale является внешним ключом, ссылающимся на ID_Client в таблице Client, тем самым указывая, какой клиент совершил такую продажу.

СУБД — это специализированное программное обеспечение, которое выступает посредником между базой данных, пользователем и приложениями. Оно предоставляет мощный инструментарий для создания, администрирования, обслуживания и использования БД. СУБД отвечает за выполнение языка SQL (или других языков запросов), управление параллельным доступом пользователей, обеспечение безопасности, поддержание целостности данных и восстановление в случае сбоев.

Для реализации проекта информационной системы отдела продаж салонов сотовой связи была выбрана Microsoft SQL Server. Этот выбор обусловлен её надёжностью, высокой производительностью, развитыми возможностями безопасности и отличной интеграцией с выбранной платформой разработки (.NET и C#), что упрощает создание слоя доступа к данным в приложении. Другие известные СУБД включают MySQL, PostgreSQL, Oracle Database, SQLite.

Таким образом, разработка базы данных включает в себя проектирование её логической структуры (на основе ER-диаграммы), создание физических таблиц, определение полей и типов данных, установление связей между таблицами с помощью первичных и внешних ключей. Заполнение БД начальными данными необходимо для проверки работоспособности системы и демонстрации её функционала.

3.3.1. Физическая ER диаграмма

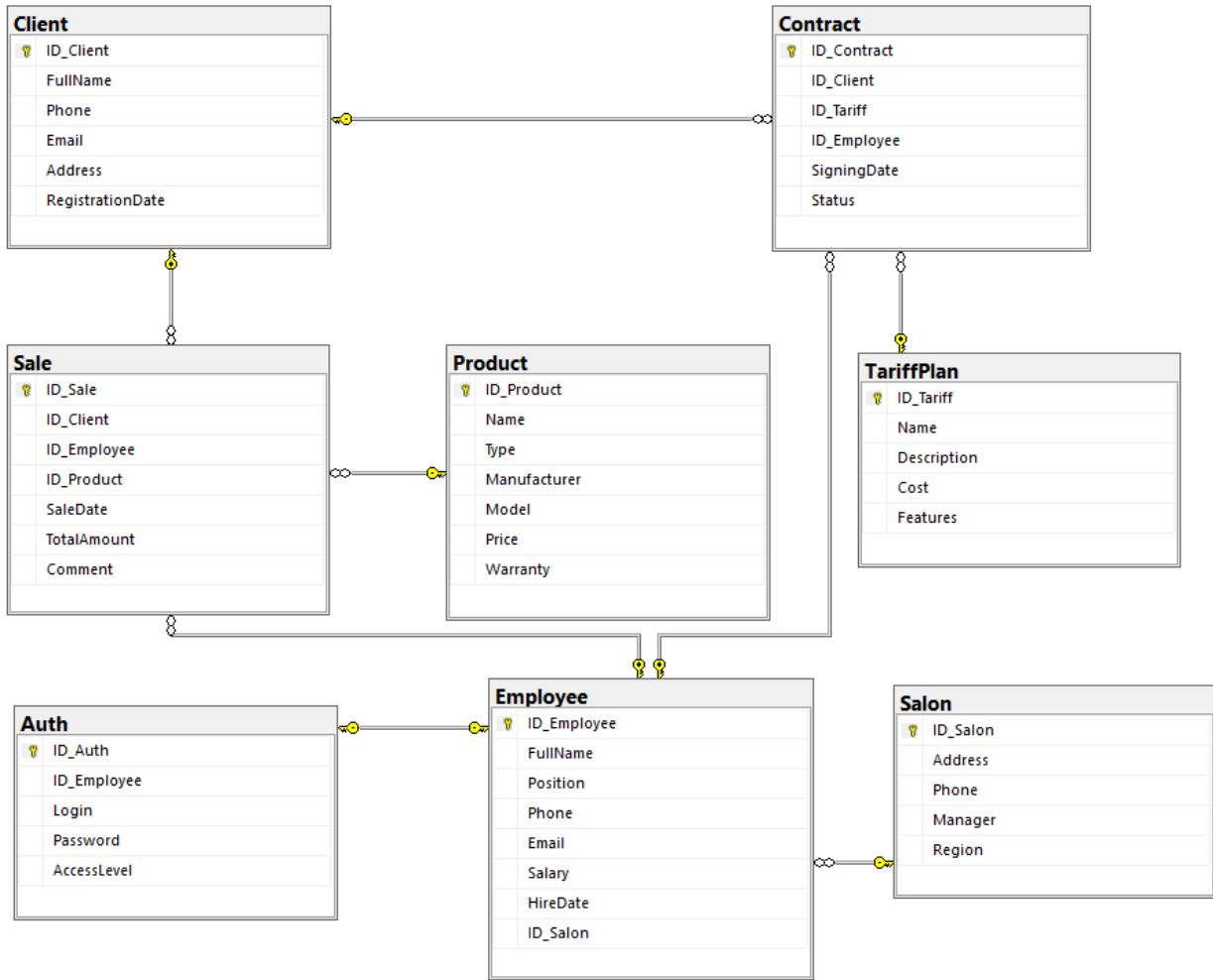


Рисунок 3.15. Физическая ER диаграмма

3.3.2. Описание таблиц

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

Описание таблиц:

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

- Phone (Телефон) — контактный номер телефона сотрудника (NVARCHAR(20), NOT NULL)
- Email (Email) — адрес электронной почты сотрудника (NVARCHAR(255), NOT NULL)
- Salary (Зарплата) — величина заработной платы (DECIMAL(10,2), NOT NULL, CHECK (Salary > 0))
- HireDate (Дата приёма) — дата, когда сотрудник был принят на работу (DATE, NOT NULL, DEFAULT GETDATE())
- ID_Salon (ID Салона) — внешний ключ, ссылающийся на таблицу Salon (INT, NOT NULL, FOREIGN KEY -> Salon.ID_Salon)

	Имя столбца	Тип данных	Разрешить знач...
	ID_Employee	int	<input type="checkbox"/>
	FullName	nvarchar(100)	<input type="checkbox"/>
	Position	nvarchar(100)	<input type="checkbox"/>
	Phone	nvarchar(20)	<input type="checkbox"/>
	Email	nvarchar(255)	<input type="checkbox"/>
	Salary	decimal(10, 2)	<input type="checkbox"/>
	HireDate	date	<input type="checkbox"/>
	ID_Salon	int	<input type="checkbox"/>

Рисунок 3.16 Таблица сотрудники

	ID_Employee	FullName	Position	Phone	Email	Salary	HireDate	ID_Salon
1	1	Смирнова Анна Петровна	Продавец-консультант	+79151234567	smimova@example.com	50000.00	2022-01-15	1
2	2	Васильев Дмитрий Иванович	Старший продавец	+79151234568	vasiliev@example.com	55000.00	2021-11-20	1
3	3	Ковалева Мария Алексеевна	Продавец-консультант	+79151234569	kovalyova@example.com	48000.00	2023-03-10	1
4	4	Морозов Иван Семенович	Продавец-консультант	+79151234570	morozov@example.com	47000.00	2023-05-05	1
5	5	Петрова Елена Николаевна	Продавец-консультант	+79151234571	petrova@example.com	49000.00	2022-09-12	1
6	6	Соколов Алексей Владимирович	Продавец-консультант	+79151234572	sokolov@example.com	46000.00	2023-07-18	1
7	7	Волкова Ольга Юрьевна	Продавец-консультант	+79151234573	volkova@example.com	48500.00	2022-12-22	1
8	8	Зайцев Николай Петрович	Продавец-консультант	+79151234574	zaytsev@example.com	47500.00	2023-01-30	1
9	9	Андреева Татьяна Дмитриевна	Продавец-консультант	+79151234575	andreeva@example.com	49500.00	2022-06-14	1
10	10	Борисов Михаил Александрович	Продавец-консультант	+79151234576	borisov@example.com	46500.00	2023-04-25	1
11	11	Михайлова Светлана Ивановна	Продавец-консультант	+79151234577	mikhaylova@example.com	48000.00	2022-10-08	1
12	12	Федоров Александр Сергеевич	Продавец-консультант	+79151234578	fedorov@example.com	47000.00	2023-08-17	1
13	13	Козлова Анастасия Викторовна	Продавец-консультант	+79151234579	kozlova@example.com	49000.00	2022-02-28	1
14	14	Павлов Дмитрий Алексеевич	Продавец-консультант	+79151234580	pavlov@example.com	46000.00	2023-06-11	1
15	15	Новикова Ирина Владимировна	Продавец-консультант	+79151234581	novikova@example.com	48500.00	2022-07-19	1
16	16	Марков Сергей Петрович	Продавец-консультант	+79151234582	markov@example.com	47500.00	2023-02-14	1
17	17	Дмитриева Екатерина Александровна	Продавец-консультант	+79151234583	dmitrieva@example.com	49500.00	2022-04-05	1
18	18	Егоров Артем Викторович	Продавец-консультант	+79151234584	egorov@example.com	46500.00	2023-09-20	1
19	19	Григорьева Наталья Ивановна	Продавец-консультант	+79151234585	grigoryeva@example.com	48000.00	2022-08-30	1
20	20	Мельников Виктор Алексеевич	Продавец-консультант	+79151234586	melnikov@example.com	47000.00	2023-03-22	1
21	21	Беляева Полина Дмитриевна	Продавец-консультант	+79151234587	belyaeva@example.com	49000.00	2022-11-15	1
22	22	Титов Игорь Сергеевич	Продавец-консультант	+79151234588	titov@example.com	46000.00	2023-10-10	1
23	23	Фролова Виктория Александровна	Продавец-консультант	+79151234589	frolova@example.com	48500.00	2022-05-18	1
24	24	Куликов Андрей Петрович	Продавец-консультант	+79151234590	kulikov@example.com	47500.00	2023-01-25	1
25	25	Макарова Дарья Ивановна	Продавец-консультант	+79151234591	makarova@example.com	49500.00	2022-09-07	1
26	26	Гусев Алексей Викторович	Продавец-консультант	+79151234592	gusev@example.com	46500.00	2023-07-03	1
27	27	Иванова Маргарита Алексеевна	Продавец-консультант	+79151234593	ivanova@example.com	48000.00	2022-12-12	1
28	28	Назаров Роман Петрович	Продавец-консультант	+79151234594	nazarov@example.com	47000.00	2023-04-15	1
29	29	Ефимова Анна Ивановна	Продавец-консультант	+79151234595	efimova@example.com	49000.00	2022-06-20	1
30	30	Гаврилов Валерий Дмитриевич	Продавец-консультант	+79151234596	gavrilov@example.com	46000.00	2023-08-05	1

Рисунок 3.17 Данные в таблице сотрудники

2) Салон (Salon)

- ID_Salon (ID) — уникальный идентификатор салона (INT, IDENTITY, PRIMARY KEY, NOT NULL)
- Address (Адрес) — полный адрес салона (NVARCHAR(255), NOT NULL)
- Phone (Телефон) — контактный номер телефона салона (NVARCHAR(20), NOT NULL)
- Manager (Руководитель) — ФИО руководителя салона (NVARCHAR(100), NOT NULL)
- Region (Регион) — регион, в котором находится салон (NVARCHAR(100), NOT NULL)

	Имя столбца	Тип данных	Разрешить знач...
	ID_Salon	int	<input type="checkbox"/>
	Address	nvarchar(255)	<input type="checkbox"/>
	Phone	nvarchar(20)	<input type="checkbox"/>
	Manager	nvarchar(100)	<input type="checkbox"/>
	Region	nvarchar(100)	<input type="checkbox"/>

Рисунок 3.18 Таблица салоны

	ID_Salon	Address	Phone	Manager	Region
1	1	ул. Пенина, д. 11, Москва	+74951234567	Иванов И.И.	Москва
2	2	пр. Мира, д. 25, Санкт-Петербург	+78121234567	Петров П.П.	Санкт-Петербург
3	3	ул. Гагарина, д. 5, Новосибирск	+73831234567	Сидоров С.С.	Новосибирск
4	4	ул. Кирова, д. 15, Екатеринбург	+73431234567	Козлов К.К.	Свердловская обл.
5	5	пр. Ленина, д. 30, Казань	+78431234567	Морозов М.М.	Татарстан
6	6	ул. Советская, д. 40, Нижний Новгород	+78311234567	Волков В.В.	Нижегородская обл.
7	7	ул. Пушкина, д. 50, Краснодар	+78611234567	Никитин Н.Н.	Краснодарский край
8	8	ул. Пермитова, д. 60, Воронеж	+74731234567	Дмитриев Д.Д.	Воронежская обл.
9	9	ул. Чехова, д. 70, Ростов-на-Дону	+78631234567	Федоров Ф.Ф.	Ростовская обл.
10	10	ул. Победы, д. 80, Самара	+78461234567	Беляев Б.Б.	Самарская обл.
11	11	ул. Маяковского, д. 90, Уфа	+73471234567	Зайцев З.З.	Башкортостан
12	12	ул. Толстого, д. 100, Пермь	+73421234567	Яковлев Я.Я.	Пермский край
13	13	ул. Гоголя, д. 110, Волгоград	+78441234567	Сорокин С.С.	Волгоградская обл.
14	14	ул. Некрасова, д. 120, Красноярск	+73911234567	Григорьев Г.Г.	Красноярский край
15	15	ул. Пушкина, д. 130, Омск	+73811234567	Лазарев П.П.	Омская обл.
16	16	ул. Ленина, д. 140, Челябинск	+73511234567	Белов Б.Б.	Челябинская обл.
17	17	ул. Мира, д. 150, Рязань	+74911234567	Комаров К.К.	Рязанская обл.
18	18	ул. Советская, д. 160, Тула	+74871234567	Орлов О.О.	Тульская обл.
19	19	ул. Космонавтов, д. 170, Ижевск	+73411234567	Андреев А.А.	Удмуртия
20	20	ул. Гагарина, д. 180, Йошкар-Ола	+78361234567	Калинин К.К.	Марий Эл
21	21	ул. Пермитова, д. 190, Саратов	+78451234567	Медведев М.М.	Саратовская обл.
22	22	ул. Победы, д. 200, Тверь	+74821234567	Алексеев А.А.	Тверская обл.
23	23	ул. Чехова, д. 210, Курск	+74711234567	Ефимов Е.Е.	Курская обл.
24	24	ул. Некрасова, д. 220, Калуга	+74841234567	Фролов Ф.Ф.	Калужская обл.
25	25	ул. Толстого, д. 230, Брянск	+74831234567	Максимов М.М.	Брянская обл.
26	26	ул. Пушкина, д. 240, Ярославль	+74851234567	Афанасьев А.А.	Ярославская обл.
27	27	ул. Гоголя, д. 250, Владимир	+74921234567	Савельев С.С.	Владimirская обл.
28	28	ул. Маяковского, д. 260, Архангельск	+78181234567	Кузнецов К.К.	Архангельская обл.
29	29	ул. Пенина, д. 270, Вологда	+78171234567	Кузьмин К.К.	Вологодская обл.
30	30	ул. Мира, д. 280, Казань	+78431234568	Макаров М.М.	Татарстан

Рисунок 3.19 Данные в таблице салоны

3) Клиенты (Clients)

- ID_Client (ID) — уникальный идентификатор клиента (INT, IDENTITY, PRIMARY KEY, NOT NULL)

- FullName (Полное имя) — ФИО клиента (NVARCHAR(100), NOT NULL)
- Phone (Телефон) — контактный номер телефона клиента (NVARCHAR(20), NOT NULL)
- Email (Email) — адрес электронной почты клиента (NVARCHAR(255), NOT NULL)
- Address (Адрес) — адрес клиента (NVARCHAR(255), NOT NULL)
- RegistrationDate (Дата регистрации) — дата регистрации клиента в системе (DATE, NOT NULL, DEFAULT GETDATE())

	Имя столбца	Тип данных	Разрешить знач...
!	ID_Client	int	<input type="checkbox"/>
	FullName	nvarchar(100)	<input type="checkbox"/>
	Phone	nvarchar(20)	<input type="checkbox"/>
	Email	nvarchar(255)	<input type="checkbox"/>
	Address	nvarchar(255)	<input type="checkbox"/>
	RegistrationDate	date	<input type="checkbox"/>

Рисунок 3.20 Таблица клиенты

	ID_Cli...	FullName	Phone	Email	Address	Registration...
1	1	Кузнецов Александр Сергеевич	+79251234567	kuznetsov@example.com	ул. Пушкина, д. 1, Москва	2022-01-10
2	2	Лебедева Ольга Петровна	+79251234568	lebedeva@example.com	пр. Мира, д. 2, Санкт-Петербург	2022-02-15
3	3	Новиков Дмитрий Иванович	+79251234569	novikov@example.com	ул. Гагарина, д. 3, Новосибирск	2022-03-20
4	4	Морозова Мария Алексеевна	+79251234570	morozova@example.com	ул. Кирова, д. 4, Екатеринбург	2022-04-25
5	5	Федоров Иван Семенович	+79251234571	fedorov@example.com	пр. Ленина, д. 5, Казань	2022-05-30
6	6	Волкова Елена Николаевна	+79251234572	volkova@example.com	ул. Советская, д. 6, Нижний Новгород	2022-06-05
7	7	Алексеев Алексей Владимирович	+79251234573	alekseev@example.com	ул. Пушкина, д. 7, Краснодар	2022-07-10
8	8	Орлова Татьяна Дмитриевна	+79251234574	orlova@example.com	ул. Пермитонова, д. 8, Воронеж	2022-08-15
9	9	Комаров Михаил Александрович	+79251234575	komarov@example.com	ул. Чехова, д. 9, Ростов-на-Дону	2022-09-20
10	10	Киселева Светлана Ивановна	+79251234576	kiseleva@example.com	ул. Победы, д. 10, Самара	2022-10-25
11	11	Макаров Александр Сергеевич	+79251234577	makarov@example.com	ул. Маяковского, д. 11, Уфа	2022-11-30
12	12	Андреева Ольга Петровна	+79251234578	andreeva@example.com	ул. Толстого, д. 12, Пермь	2022-12-05
13	13	Ковалев Дмитрий Иванович	+79251234579	kovalyov@example.com	ул. Гоголя, д. 13, Волгоград	2023-01-10
14	14	Иванова Мария Алексеевна	+79251234580	ivanova@example.com	ул. Некрасова, д. 14, Красноярск	2023-02-15
15	15	Григорьев Иван Семенович	+79251234581	grigoryov@example.com	ул. Пушкина, д. 15, Омск	2023-03-20
16	16	Беляева Елена Николаевна	+79251234582	belyaeva@example.com	ул. Ленина, д. 16, Челябинск	2023-04-25
17	17	Зайцев Алексей Владимирович	+79251234583	zaytsev@example.com	ул. Мира, д. 17, Рязань	2023-05-30
18	18	Соколова Татьяна Дмитриевна	+79251234584	sokolova@example.com	ул. Советская, д. 18, Тула	2023-06-05
19	19	Яковлева Михаил Александрович	+79251234585	yakovleva@example.com	ул. Космонавтов, д. 19, Ижевск	2023-07-10
20	20	Медведев Светлана Ивановна	+79251234586	medvedev@example.com	ул. Гагарина, д. 20, Йошкар-Ола	2023-08-15
21	21	Афанасьева Александр Сергеев...	+79251234587	afanasieva@example.c...	ул. Пермитонова, д. 21, Саратов	2023-09-20
22	22	Савельев Ольга Петровна	+79251234588	saveliev@example.com	ул. Победы, д. 22, Тверь	2023-10-25
23	23	Кузьмин Дмитрий Иванович	+79251234589	kuzmin@example.com	ул. Чехова, д. 23, Курск	2023-11-30
24	24	Калинина Мария Алексеевна	+79251234590	kalinina@example.com	ул. Некрасова, д. 24, Калуга	2023-12-05
25	25	Максимов Иван Семенович	+79251234591	maksimov@example.com	ул. Толстого, д. 25, Брянск	2024-01-10
26	26	Фролова Елена Николаевна	+79251234592	frolova@example.com	ул. Пушкина, д. 26, Ярославль	2024-02-15
27	27	Ефимов Алексей Владимирович	+79251234593	efimov@example.com	ул. Гоголя, д. 27, Владимир	2024-03-20
28	28	Александрова Татьяна Дмитри...	+79251234594	aleksandrova@example...	ул. Маяковского, д. 28, Архангельск	2024-04-25
29	29	Семенов Михаил Александрович	+79251234595	semenov@example.com	ул. Ленина, д. 29, Вологда	2024-05-30
30	30	Миронова Светлана Ивановна	+79251234596	mironova@example.com	ул. Мира, д. 30, Казань	2024-06-05

Рисунок 3.21 Данные в таблице клиенты

4) Товары (Products)

- ID_Product (ID) — уникальный идентификатор товара (INT, IDENTITY, PRIMARY KEY, NOT NULL)
- Name (Название) — наименование товара (NVARCHAR(255), NOT NULL)
 - Type (Тип) — категория товара (например, 'Телефон', 'Аксессуар') (NVARCHAR(50), NOT NULL)
 - Manufacturer (Производитель) — название компании-производителя (NVARCHAR(100), NOT NULL)
 - Model (Модель) — модель товара (NVARCHAR(100), NULLABLE)
 - Price (Цена) — цена товара (DECIMAL(10,2), NOT NULL, CHECK (Price >= 0))
 - Warranty (Гарантия) — описание гарантийных условий (NVARCHAR(50), NULLABLE)

	Имя столбца	Тип данных	Разрешить знач...
1	ID_Product	int	<input type="checkbox"/>
	Name	nvarchar(255)	<input type="checkbox"/>
	Type	nvarchar(50)	<input type="checkbox"/>
	Manufacturer	nvarchar(100)	<input type="checkbox"/>
	Model	nvarchar(100)	<input checked="" type="checkbox"/>
	Price	decimal(10, 2)	<input type="checkbox"/>
	Warranty	nvarchar(50)	<input checked="" type="checkbox"/>

Рисунок 3.22 Таблица товары

	ID_Product	Name	Type	Manufacturer	Model	Price	Warranty
1	1	iPhone 15 Pro	Телефон	Apple	iPhone 15 Pro	110000.00	12 месяцев
2	2	Samsung Galaxy S24	Телефон	Samsung	Galaxy S24	95000.00	12 месяцев
3	3	Xiaomi 14	Телефон	Xiaomi	14	65000.00	12 месяцев
4	4	Huawei P60	Телефон	Huawei	P60	75000.00	12 месяцев
5	5	OnePlus 12	Телефон	OnePlus	12	85000.00	12 месяцев
6	6	Google Pixel 8	Телефон	Google	Pixel 8	90000.00	12 месяцев
7	7	Sony Xperia 1 V	Телефон	Sony	Xperia 1 V	110000.00	12 месяцев
8	8	Motorola Edge 40	Телефон	Motorola	Edge 40	55000.00	12 месяцев
9	9	Nokia G42	Телефон	Nokia	G42	25000.00	12 месяцев
10	10	Realme GT Neo 5	Телефон	Realme	GT Neo 5	45000.00	12 месяцев
11	11	OPPO Find X6	Телефон	OPPO	Find X6	80000.00	12 месяцев
12	12	Vivo X Flip	Телефон	Vivo	X Flip	100000.00	12 месяцев
13	13	ASUS ROG Phone 7	Телефон	ASUS	ROG Phone 7	130000.00	12 месяцев
14	14	BlackBerry Key2	Телефон	BlackBerry	Key2	70000.00	12 месяцев
15	15	LG Velvet	Телефон	LG	Velvet	60000.00	12 месяцев
16	16	ZTE Axon 40 Ultra	Телефон	ZTE	Axon 40 Ultra	95000.00	12 месяцев
17	17	Honor Magic 5	Телефон	Honor	Magic 5	85000.00	12 месяцев
18	18	Nothing Phone (2)	Телефон	Nothing	Phone (2)	75000.00	12 месяцев
19	19	Fairphone 5	Телефон	Fairphone	5	65000.00	12 месяцев
20	20	CAT S62 Pro	Телефон	CAT	S62 Pro	70000.00	12 месяцев
21	21	Кабель USB-C	Аксесс...	Generic	USB-C 1m	500.00	6 месяцев
22	22	Кабель Lightning	Аксесс...	Apple	Lightning 1m	2000.00	6 месяцев
23	23	Наушники Bluetooth	Аксесс...	Sony	WH-CH720N	8000.00	12 месяцев
24	24	Чехол для iPhone ...	Аксесс...	UAG	Meteor	4000.00	6 месяцев
25	25	Чехол для Samsun...	Аксесс...	Spigen	Tough Armor	2500.00	6 месяцев
26	26	Пленка защитная	Аксесс...	Tempered ...	Full Cover	800.00	Нет
27	27	Зарядное устройс...	Аксесс...	Anker	PowerPort II...	3000.00	12 месяцев
28	28	Портативный акк...	Аксесс...	RAVPower	10000mAh	4500.00	12 месяцев
29	29	Карта памяти 128...	Аксесс...	Samsung	EVO Plus	2000.00	12 месяцев
30	30	Чехол-книжка	Аксесс...	Olixar	Pro	3500.00	6 месяцев

Рисунок 3.23 Данные в таблице торары

5) Тарифные планы (TariffPlans)

- ID_Tariff (ID) — уникальный идентификатор тарифного плана (INT, IDENTITY, PRIMARY KEY, NOT NULL)
 - Name (Название) — наименование тарифного плана (NVARCHAR(100), NOT NULL)

- Description (Описание) — подробное описание тарифа (NVARCHAR(MAX), NOT NULL)
- Cost (Стоимость) — стоимость тарифного плана (DECIMAL(10,2), NOT NULL, CHECK (Cost >= 0))
- Features (Особенности) — уникальные особенности тарифа (NVARCHAR(MAX), NULLABLE)

	Имя столбца	Тип данных	Разрешить знач...
1	ID_Tariff	int	<input type="checkbox"/>
2	Name	nvarchar(100)	<input type="checkbox"/>
3	Description	nvarchar(MAX)	<input type="checkbox"/>
4	Cost	decimal(10, 2)	<input type="checkbox"/>
5	Features	nvarchar(MAX)	<input checked="" type="checkbox"/>

Рисунок 3.24 Таблица тарифные планы

	ID_Tariff	Name	Description	Cost	Features
1	1	Тариф "Общайся"	Безлимитные звонки внутри сети, 10 ГБ интернета	500.00	Безлимит, 10 ГБ
2	2	Тариф "Интернет"	Безлимитный интернет, 500 минут	600.00	Безлимит интернет, 500 мин
3	3	Тариф "Мегафон"	Безлимит на всё, 30 ГБ интернета	1000.00	Безлимит всё, 30 ГБ
4	4	Тариф "Смарт"	1000 минут, 20 ГБ интернета	700.00	1000 мин, 20 ГБ
5	5	Тариф "Для семьи"	Объединённый тариф для 4 человек	2000.00	4 SIM, безлимит
6	6	Тариф "Мини"	50 минут, 1 ГБ интернета	150.00	50 мин, 1 ГБ
7	7	Тариф "Бизнес"	Неограниченные звонки, 50 ГБ интернета	1500.00	Безлимит, 50 ГБ
8	8	Тариф "Детский"	Контролируемый тариф	200.00	Контроль, 2 ГБ
9	9	Тариф "Тревел"	Для поездок по миру	1200.00	Роуминг, 15 ГБ
10	10	Тариф "VIP"	Премиум тариф с бонусами	2500.00	Бонусы, 100 ГБ
11	11	Тариф "Ноль"	Плата за каждую минуту и МБ	0.00	По тарифу
12	12	Тариф "Учебный"	Скидка для студентов	400.00	Скидка 20%, 15 ГБ
13	13	Тариф "Спорт"	Бонусы для спортивных приложений	800.00	Бонусы, 25 ГБ
14	14	Тариф "Музыка"	Безлимит на музыкальные сервисы	900.00	Безлимит музыка, 30 ГБ
15	15	Тариф "Игровой"	Безлимит на игровые сервисы	1100.00	Безлимит игры, 35 ГБ
16	16	Тариф "Ночной"	Скидка наочные звонки и интернет	550.00	Ночная скидка, 12 ГБ
17	17	Тариф "Почта"	Безлимит на почтовые сервисы	650.00	Безлимит почта, 18 ГБ
18	18	Тариф "Облако"	Безлимит на облачные сервисы	750.00	Безлимит облако, 22 ГБ
19	19	Тариф "Фото"	Безлимит на фото- и видео-сервисы	850.00	Безлимит фото, 28 ГБ
20	20	Тариф "Финансы"	Безлимит на финансовые приложения	950.00	Безлимит финансы, 32 ГБ
21	21	Тариф "Медицина"	Безлимит на медицинские приложения	1050.00	Безлимит медицина, 38 ГБ
22	22	Тариф "Образова...	Безлимит на образовательные ресурсы	1150.00	Безлимит образование, 42...
23	23	Тариф "Транспорт"	Безлимит на транспортные приложения	1250.00	Безлимит транспорт, 45 ГБ
24	24	Тариф "Покупки"	Безлимит на торговые площадки	1350.00	Безлимит покупки, 48 ГБ
25	25	Тариф "Развлече...	Безлимит на развлекательные сервисы	1450.00	Безлимит развлечения, 52...
26	26	Тариф "Новости"	Безлимит на новостные ресурсы	1550.00	Безлимит новости, 55 ГБ
27	27	Тариф "Погода"	Безлимит на метеорологические сервисы	1650.00	Безлимит погода, 58 ГБ
28	28	Тариф "Путешест...	Безлимит на туристические сервисы	1750.00	Безлимит путешествия, 62...
29	29	Тариф "Еда"	Безлимит на доставку еды	1850.00	Безлимит еда, 65 ГБ
30	30	Тариф "Социальн...	Безлимит на соцсети	1950.00	Безлимит соцсети, 70 ГБ

Рисунок 3.25 Данные в таблице тарифные планы

6) Продажи (Sales)

- ID_Sale (ID) — уникальный идентификатор продажи (INT, IDENTITY, PRIMARY KEY, NOT NULL)
 - ID_Client (ID Клиента) — внешний ключ, ссылающийся на таблицу Clients (INT, NOT NULL, FOREIGN KEY -> Clients.ID_Client)
 - ID_Employee (ID Сотрудника) — внешний ключ, ссылающийся на таблицу Employees (INT, NOT NULL, FOREIGN KEY -> Employees.ID_Employee)
 - ID_Product (ID Товара) — внешний ключ, ссылающийся на таблицу Products (INT, NOT NULL, FOREIGN KEY -> Products.ID_Product)
 - SaleDate (Дата продажи) — дата и время оформления продажи (DATETIME2, NOT NULL, DEFAULT GETDATE())
 - TotalAmount (Сумма) — общая сумма продажи (DECIMAL(10,2), NOT NULL, CHECK (TotalAmount >= 0))
 - Comment (Комментарий) — дополнительная информация по продаже (NVARCHAR(MAX), NULLABLE)

	Имя столбца	Тип данных	Разрешить знач...
!	ID_Sale	int	<input type="checkbox"/>
	ID_Client	int	<input checked="" type="checkbox"/>
	ID_Employee	int	<input type="checkbox"/>
	ID_Product	int	<input type="checkbox"/>
	SaleDate	datetime2(7)	<input type="checkbox"/>
	TotalAmount	decimal(10, 2)	<input type="checkbox"/>
	Comment	nvarchar(MAX)	<input checked="" type="checkbox"/>

Рисунок 3.26 Таблица продажи

	ID_Sale	ID_Client	ID_Employee	ID_Product	SaleDate	TotalAmount	Comment
1	1	1	1	1	2025-11-15 10:30:00.0000000	120000.00	Покупка в рассрочку
2	2	2	2	2	2025-11-15 11:15:00.0000000	95000.00	Без комментариев
3	3	3	3	3	2025-11-15 12:00:00.0000000	65000.00	Подарок
4	4	4	4	4	2025-11-15 13:30:00.0000000	75000.00	Обмен старого телефона
5	5	5	5	5	2025-11-15 14:45:00.0000000	85000.00	Без комментариев
6	6	6	6	6	2025-11-15 09:15:00.0000000	90000.00	Без комментариев
7	7	7	7	7	2025-10-15 10:00:00.0000000	110000.00	Покупка с тарифом
8	8	8	8	8	2025-10-15 11:30:00.0000000	55000.00	Без комментариев
9	9	9	9	9	2025-10-15 12:45:00.0000000	25000.00	Без комментариев
10	10	10	10	10	2025-10-15 14:00:00.0000000	45000.00	Без комментариев
11	11	11	11	11	2025-10-15 09:30:00.0000000	80000.00	Без комментариев
12	12	12	12	12	2025-10-15 10:15:00.0000000	100000.00	Без комментариев
13	13	13	13	13	2025-10-15 11:00:00.0000000	130000.00	Без комментариев
14	14	14	14	14	2025-10-15 12:30:00.0000000	70000.00	Без комментариев
15	15	15	15	15	2025-10-15 13:45:00.0000000	60000.00	Без комментариев
16	16	16	16	16	2025-10-15 09:00:00.0000000	95000.00	Без комментариев
17	17	17	17	17	2025-10-15 10:30:00.0000000	85000.00	Без комментариев
18	18	18	18	18	2024-01-18 11:15:00.0000000	75000.00	Без комментариев
19	19	19	19	19	2024-01-18 12:00:00.0000000	65000.00	Без комментариев
20	20	20	20	20	2024-01-18 13:30:00.0000000	70000.00	Без комментариев
21	21	21	21	21	2024-01-19 09:45:00.0000000	500.00	Доп. аксессуар
22	22	22	22	22	2024-01-19 10:00:00.0000000	2000.00	Доп. аксессуар
23	23	23	23	23	2024-01-19 11:30:00.0000000	8000.00	Доп. аксессуар
24	24	24	24	24	2024-01-19 12:15:00.0000000	4000.00	Доп. аксессуар
25	25	25	25	25	2024-01-19 13:00:00.0000000	2500.00	Доп. аксессуар
26	26	26	26	26	2024-01-20 09:15:00.0000000	800.00	Доп. аксессуар
27	27	27	27	27	2024-01-20 10:00:00.0000000	3000.00	Доп. аксессуар
28	28	28	28	28	2024-01-20 11:30:00.0000000	4500.00	Доп. аксессуар
29	29	29	29	29	2024-01-20 12:45:00.0000000	2000.00	Доп. аксессуар
30	30	30	30	30	2024-01-20 14:00:00.0000000	3500.00	Доп. аксессуар
31	32	2	4	5	2025-11-18 17:51:44.1433333	85000.00	1
32	33	4	1	1	2025-11-18 17:57:29.0966667	120000.00	
33	34	NULL	1	12	2025-11-18 17:59:48.2800000	100000.00	
34	37	NULL	1	7	2025-11-18 18:03:18.7233333	110000.00	
35	38	NULL	1	1	2025-11-18 18:03:22.0100000	120000.00	
36	39	NULL	1	12	2025-11-18 18:03:24.7600000	100000.00	
37	40	NULL	1	2	2025-11-18 18:03:27.7800000	95000.00	
38	41	NULL	1	21	2025-11-19 10:31:24.5333333	500.00	

Рисунок 3.27 Данные в таблице продажи

7) Контракты (Contracts)

- ID_Contract (ID) — уникальный идентификатор контракта (INT, IDENTITY, PRIMARY KEY, NOT NULL)
- ID_Client (ID Клиента) — внешний ключ, ссылающийся на таблицу Clients (INT, NOT NULL, FOREIGN KEY -> Clients.ID_Client)
- ID_Tariff (ID Тарифа) — внешний ключ, ссылающийся на таблицу TariffPlans (INT, NOT NULL, FOREIGN KEY -> TariffPlans.ID_Tariff)

- ID_Employee (ID Сотрудника) — внешний ключ, ссылающийся на таблицу Employees (INT, NOT NULL, FOREIGN KEY -> Employees.ID_Employee)
- SigningDate (Дата заключения) — дата заключения контракта (DATE, NOT NULL, DEFAULT GETDATE())
- Status (Статус) — текущий статус контракта (например, 'Активен', 'Завершен') (NVARCHAR(50), NOT NULL)

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID_Contract	int	<input type="checkbox"/>
	ID_Client	int	<input type="checkbox"/>
	ID_Tariff	int	<input type="checkbox"/>
	ID_Employee	int	<input type="checkbox"/>
	SigningDate	date	<input type="checkbox"/>
	Status	nvarchar(50)	<input type="checkbox"/>

Рисунок 3.28 Таблица контракты

	ID_Contract	ID_Client	ID_Tariff	ID_Employee	SigningDate	Status
1	1	1	1	1	2024-01-15	Активен
2	2	2	2	1	2024-01-15	Активен
3	3	3	3	3	2024-01-15	Активен
4	4	4	4	4	2024-01-15	Активен
5	5	5	5	5	2024-01-16	Активен
6	6	6	6	6	2024-01-16	Активен
7	7	7	7	7	2024-01-16	Активен
8	8	8	8	8	2024-01-16	Активен
9	9	9	9	9	2024-01-17	Активен
10	10	10	10	10	2024-01-17	Активен
11	11	11	11	11	2024-01-17	Активен
12	12	12	12	12	2024-01-17	Активен
13	13	13	13	13	2024-01-18	Активен
14	14	14	14	14	2024-01-18	Активен
15	15	15	15	15	2024-01-18	Активен
16	16	16	16	16	2024-01-18	Активен
17	17	17	17	17	2024-01-19	Активен
18	18	18	18	18	2024-01-19	Активен
19	19	19	19	19	2024-01-19	Активен
20	20	20	20	20	2024-01-19	Активен
21	21	21	21	21	2024-01-20	Активен
22	22	22	22	22	2024-01-20	Активен
23	23	23	23	23	2024-01-20	Активен
24	24	24	24	24	2024-01-20	Активен
25	25	25	25	25	2024-01-21	Активен
26	26	26	26	26	2024-01-21	Активен
27	27	27	27	27	2024-01-21	Активен
28	28	28	28	28	2024-01-21	Активен
29	29	29	29	29	2024-01-22	Активен
30	30	30	30	30	2024-01-22	Активен
31	32	14	16	2	2025-11-18	111

Рисунок 3.29 Данные в таблице контракты

8) Авторизация (Auth)

- ID_Auth (ID) — уникальный идентификатор записи авторизации (INT, IDENTITY, PRIMARY KEY, NOT NULL)
 - ID_Employee (ID Сотрудника) — внешний ключ, ссылающийся на таблицу Employees, одновременно является уникальным (UNIQUE) (INT, NOT NULL, FOREIGN KEY -> Employees.ID_Employee)
 - Login (Логин) — уникальное имя пользователя для входа в систему (NVARCHAR(50), NOT NULL, UNIQUE)

- Password (Пароль) — пароль пользователя (NVARCHAR(255), NOT NULL)
- AccessLevel (Уровень доступа) — уровень доступа пользователя (роль) (NVARCHAR(50), NOT NULL)

	Имя столбца	Тип данных	Разрешить знач...
1	ID_Auth	int	<input type="checkbox"/>
	ID_Employee	int	<input type="checkbox"/>
	Login	nvarchar(50)	<input type="checkbox"/>
	Password	nvarchar(255)	<input type="checkbox"/>
	AccessLevel	nvarchar(50)	<input type="checkbox"/>

Рисунок 3.30 Таблица авторизация

	ID_Auth	ID_Employee	Login	Password	AccessLevel
1	1	1	user	pass	Admin
2	2	2	user2	password2	Manager
3	3	3	user3	password3	HeadManager
4	4	4	user4	password4	Not
5	5	5	user5	password5	Not
6	6	6	user6	password6	Not
7	7	7	user7	password7	Not
8	8	8	user8	password8	Not
9	9	9	user9	password9	Not
10	10	10	user10	password10	Not
11	11	11	user11	password11	Not
12	12	12	user12	password12	Not
13	13	13	user13	password13	Not
14	14	14	user14	password14	Not
15	15	15	user15	password15	Not
16	16	16	user16	password16	Not
17	17	17	user17	password17	Not
18	18	18	user18	password18	Not
19	19	19	user19	password19	Not
20	20	20	user20	password20	Not
21	21	21	user21	password21	Not
22	22	22	user22	password22	Not
23	23	23	user23	password23	Not
24	24	24	user24	password24	Not
25	25	25	user25	password25	Not
26	26	26	user26	password26	Not
27	27	27	user27	password27	Not
28	28	28	user28	password28	Not
29	29	29	user29	password29	Not
30	30	30	user30	password30	Not

Рисунок 3.31 Данные в таблице авторизация

3.3.3. Связи между таблицами

База данных информационной системы отдела продаж салонов сотовой связи спроектирована как реляционная, где таблицы связаны между собой с помощью внешних ключей. Это обеспечивает целостность данных и логическую связь между различными сущностями, такими как клиенты, сотрудники, товары, продажи и контракты. Ниже приведено описание основных связей между таблицами.

- 1) Сотрудники (Employees) → Салон (Salon)
 - Тип связи: Один ко многим (1:N)
 - Описание: Один салон может иметь множество сотрудников.
 - Реализация: В таблице Employees поле ID_Salon является внешним ключом, который ссылается на первичный ключ ID_Salon в таблице Salon.
- 2) Сотрудники (Employees) → Авторизация (Auth)
 - Тип связи: Один к одному (1:1)
 - Описание: У каждого сотрудника может быть только одна запись авторизации, и наоборот.
 - Реализация: В таблице Auth поле ID_Employee является внешним ключом, которое одновременно является уникальным (UNIQUE) и ссылается на первичный ключ ID_Employee в таблице Employees.
- 3) Продажи (Sales) → Клиенты (Clients)
 - Тип связи: Один ко многим (1:N)
 - Описание: Один клиент может совершить множество продаж.
 - Реализация: В таблице Sales поле ID_Client является внешним ключом, который ссылается на первичный ключ ID_Client в таблице Clients.
- 4) Продажи (Sales) → Сотрудники (Employees)
 - Тип связи: Один ко многим (1:N)
 - Описание: Один сотрудник может оформить множество продаж.
 - Реализация: В таблице Sales поле ID_Employee является внешним ключом, который ссылается на первичный ключ ID_Employee в таблице Employees.

5) Продажи (Sales) → Товары (Products)

- Тип связи: Один ко многим (1:N)
- Описание: Один товар может участвовать в множестве продаж (один и тот же тип товара продается разным клиентам).
 - Реализация: В таблице Sales поле ID_Product является внешним ключом, который ссылается на первичный ключ ID_Product в таблице Products.

6) Контракты (Contracts) → Клиенты (Clients)

- Тип связи: Один ко многим (1:N)
- Описание: Один клиент может заключить множество контрактов.
- Реализация: В таблице Contracts поле ID_Client является внешним ключом, который ссылается на первичный ключ ID_Client в таблице Clients.

7) Контракты (Contracts) → Тарифные планы (TariffPlans)

- Тип связи: Один ко многим (1:N)
- Описание: Один тарифный план может быть выбран множеством клиентов (заключено множество контрактов на один тариф).
- Реализация: В таблице Contracts поле ID_Tariff является внешним ключом, который ссылается на первичный ключ ID_Tariff в таблице TariffPlans.

8) Контракты (Contracts) → Сотрудники (Employees)

- Тип связи: Один ко многим (1:N)
- Описание: Один сотрудник может оформить множество контрактов.
- Реализация: В таблице Contracts поле ID_Employee является внешним ключом, который ссылается на первичный ключ ID_Employee в таблице Employees.

Эти связи обеспечивают целостность и логичность структуры базы данных, позволяя системе корректно отражать реальные бизнес-процессы отдела продаж.

3.3.4. Индексы в БД

В базе данных индексы — это специальные структуры, созданные для ускорения поиска и сортировки записей в таблицах. Их можно сравнить с оглавлением или указателем в книге: они позволяют системе оперативно находить нужные данные, не перебирая всю таблицу целиком. Индекс строится на основе одного или нескольких столбцов и включает в себя отсортированные значения этих столбцов, а также ссылки на соответствующие строки в самой таблице. Использование индексов существенно ускоряет выполнение запросов, особенно операций вида SELECT, JOIN, WHERE и ORDER BY. Вместе с тем, наличие индексов требует дополнительного места на диске и может замедлять операции добавления, изменения и удаления записей, поскольку при каждом изменении данных в таблице нужно также обновлять и индекс. Следовательно, столбцы для индексации стоит выбирать обдуманно, основываясь на реальных сценариях использования и частоте обращения к тем или иным данным.

Ниже приведён список индексов, которые будут созданы в скрипте базы данных для улучшения производительности типовых запросов:

- IX_Employee_ID_Salon (Таблица Employee): Индекс на столбце ID_Salon для ускорения поиска сотрудников по салону, в котором они работают.
- IX_Sale_ID_Client (Таблица Sale): Индекс на столбце ID_Client для ускорения поиска продаж по конкретному клиенту.
- IX_Sale_ID_Employee (Таблица Sale): Индекс на столбце ID_Employee для ускорения поиска продаж, оформленных конкретным сотрудником.
- IX_Sale_ID_Product (Таблица Sale): Индекс на столбце ID_Product для ускорения поиска продаж, в которых участвует конкретный товар.
- IX_Contract_ID_Client (Таблица Contract): Индекс на столбце ID_Client для ускорения поиска контрактов по клиенту.
- IX_Contract_ID_Tariff (Таблица Contract): Индекс на столбце ID_Tariff для ускорения поиска контрактов по тарифному плану.

- IX_Contract_ID_Employee (Таблица Contract): Индекс на столбце ID_Employee для ускорения поиска контрактов, оформленных конкретным сотрудником.
- IX_Auth_ID_Employee (Таблица Auth): Индекс на столбце ID_Employee (он же UNIQUE) для ускорения поиска данных авторизации по идентификатору сотрудника. Также обеспечивает уникальность связи между сотрудником и его учётной записью.
- IX_Client_FullName (Таблица Client): Индекс на столбце FullName для ускорения поиска клиентов по ФИО.
- IX_Client_Phone (Таблица Client): Индекс на столбце Phone для ускорения поиска клиентов по номеру телефона.
- IX_Product_Name (Таблица Product): Индекс на столбце Name для ускорения поиска товаров по названию.
- IX_Product_Type (Таблица Product): Индекс на столбце Type для ускорения поиска товаров по типу (например, "Телефон", "Аксессуар").
- IX_TariffPlan_Name (Таблица TariffPlan): Индекс на столбце Name для ускорения поиска тарифных планов по названию.
- IX_Salon_Address (Таблица Salon): Индекс на столбце Address для ускорения поиска салонов по адресу.

Использование этих индексов позволяет повысить общую производительность системы за счёт ускорения типовых операций поиска и фильтрации, что особенно важно при работе с большими объёмами данных и многопользовательским доступом.

3.4. Отладка и тестирование программы

Процесс отладки и тестирования играет критически важную роль на завершающих этапах разработки программного обеспечения. Он направлен на обеспечение надежности, корректности функционирования и соответствия создаваемой информационной системы заявленным требованиям. Основная цель данных мероприятий в рамках проекта "Информационная система отдела продаж салонов сотовой связи" заключается в выявлении и устранении

потенциальных дефектов в коде и логике приложения до его передачи в эксплуатацию. Это позволяет гарантировать, что система будет стабильно работать, выполнять все необходимые функции и эффективно поддерживать бизнес-процессы отдела продаж.

Основной задачей стало подтверждение соответствия разработанного ПО сформулированным функциональным и нефункциональным критериям, а также обеспечение его стабильной работы в различных сценариях. В ходе работы проверялась корректность выполнения всех предусмотренных функций (например, ведение справочников, оформление продаж и контрактов), взаимодействие с базой данных, корректность работы пользовательского интерфейса, соблюдение политик разграничения доступа для разных ролей, устойчивость к некорректному вводу данных и сбоям, а также оценка производительности при работе с типичными объемами информации.

Для достижения поставленной цели использовались различные подходы к тестированию. Основное внимание было уделено функциональному тестированию, которое проводилось вручную на основе заранее подготовленных сценариев. Эти сценарии охватывали ключевые действия: вход в систему, работу с данными клиентов и сотрудников, оформление продаж и контрактов, редактирование справочников, поиск и фильтрацию, а также проверку реакции системы на некорректные действия. Также проводилось регрессионное тестирование после внесения изменений, чтобы убедиться в отсутствии негативного влияния на уже протестированный функционал. Особое внимание уделялось обработке исключений и граничных условий, чтобы система оставалась устойчивой к ошибкам пользователя.

В работе активно использовались встроенные средства отладки среды Microsoft Visual Studio Community. Они позволили эффективно находить и устранять логические и иные ошибки в программе, устанавливая точки останова, отслеживая значения переменных и анализируя стек вызовов.

Тестирование проводилось поэтапно: сначала проверялись отдельные компоненты и логика, затем взаимодействие между ними, и, наконец, система в целом. Результаты тестов анализировались, выявленные ошибки документировались, устранялись, и соответствующие функции тестировались повторно. В результате была достигнута высокая степень уверенности в корректности работы основного функционала. Система показала устойчивость к стандартным сценариям использования и адекватно реагирует на большинство ошибок ввода. Таким образом, этап отладки и тестирования позволил создать надежный и качественный программный продукт, готовый к использованию.

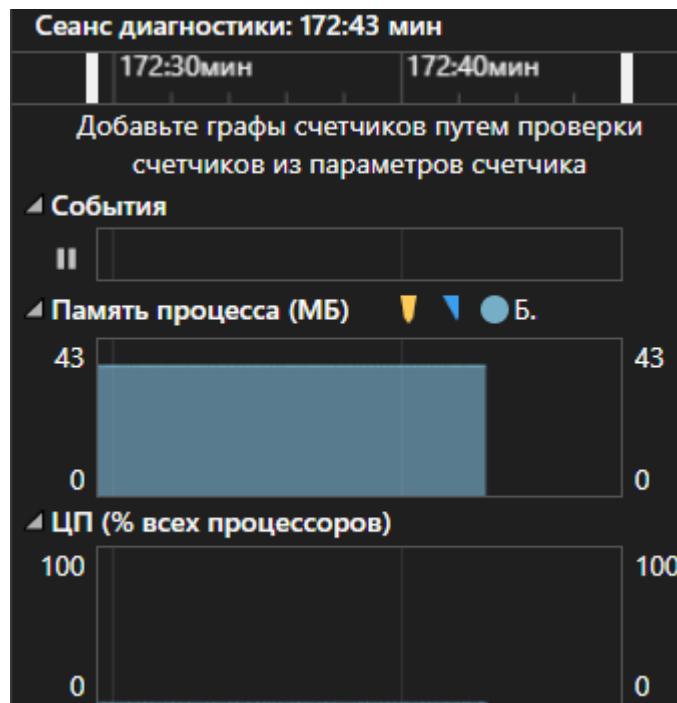


Рисунок 3.32. Производительность программы, используемые ресурсы

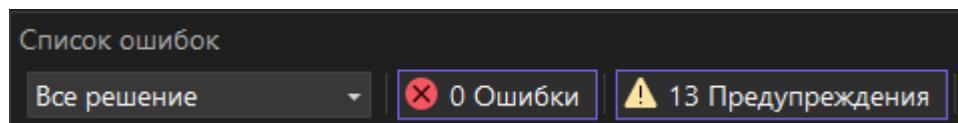


Рисунок 3.33. Список ошибок

```
92
93
94
95
96     if (obj.ID_Client != null) query.CommandText = "INSERT INTO Sale VALUES (@ID_Client, @ID_Employee, @ID_Product, @SaleDate, @TotalAmount, @Comment);";
97     else query.CommandText = "INSERT INTO Sale (ID_Employee, ID_Product, SaleDate, TotalAmount, Comment) VALUES (@ID_Employee, @ID_Product, @SaleDate, @TotalAmount, @Comment);"
98
99     if (obj.ID_Client != null) query.Parameters.AddWithValue("@ID_Client", obj.ID_Client); ≤ 1 мс прошло
100    query.Parameters.AddWithValue("@ID_Employee", obj.ID_Employee);
101    query.Parameters.AddWithValue("@ID_Product", obj.ID_Product);
102    query.Parameters.AddWithValue("@SaleDate", obj.SaleDate);
103    query.Parameters.AddWithValue("@TotalAmount", obj.TotalAmount);
104    query.Parameters.AddWithValue("@Comment", obj.Comment);
105    query.ExecuteNonQuery();
106
107  }
108
109 } catch { return -1; }
110 int _id = -1;
111 try
112 {
113     using (SqlConnection connection = new SqlConnection(SQL._sqlConnectStr))
114     {

```

Рисунок 3.34 Отладка кода (точка останова)

3.5 Руководство пользователя

Настоящее руководство пользователя предназначено для ознакомления с порядком установки, настройки и эксплуатации информационной системы отдела продаж салонов сотовой связи. Оно содержит пошаговые инструкции по работе с основными функциями системы и рассчитано на различных пользователей, обладающих разными уровнями доступа (Менеджер, Старший менеджер, Администратор).

Назначение и условия применения

Информационная система отдела продаж салонов сотовой связи разработана с целью автоматизации и оптимизации ключевых бизнес-процессов, связанных с продажей товаров и услуг, управлением клиентами, сотрудниками и аналитикой. Система предоставляет удобный интерфейс для работы с данными и позволяет различным категориям пользователей выполнять свои профессиональные задачи эффективно и с минимальными ошибками.

Система предназначена для использования в среде Microsoft Windows. Для корректной работы необходимо, чтобы на компьютере пользователя были установлены требуемые зависимости, такие как .NET Framework и драйверы для подключения к базе данных SQL Server.

Краткое описание возможностей

Система предоставляет следующие основные возможности:

- Управление клиентами: Добавление, просмотр, редактирование и поиск данных о клиентах.

- Управление товарами и тарифами: Просмотр справочников товаров и тарифных планов.
- Оформление продаж и контрактов: Создание записей о продажах товаров и заключении контрактов на услуги.
- Работа с сотрудниками: (доступно не для всех ролей) Просмотр, добавление, редактирование данных о сотрудниках.
- Управление данными: (доступно не для всех ролей) Управление справочниками (клиенты, сотрудники, товары, тарифы, салоны).
- Аналитика: (доступно Администратору) Просмотр аналитики по продажам салонов.
- Разграничение прав доступа: Возможности, доступные пользователю, зависят от его роли (Менеджер, Старший менеджер, Администратор).

Порядок запуска и авторизации

1) Запуск приложения: дважды щелкните по ярлыку приложения на рабочем столе или найдите исполняемый файл (например, MobilePhoneStoreApp.exe) и запустите его.

2) Окно авторизации: после запуска откроется окно "Авторизация".

- Введите свой Логин в соответствующее поле.
- Введите свой Пароль в соответствующее поле.
- Нажмите кнопку "Войти".

3) Проверка данных: Система проверит введенные учетные данные.

- Если логин и пароль корректны, откроется главное окно системы, соответствующее вашей роли.

- Если логин или пароль введены неверно, на экране появится сообщение об ошибке. Повторите ввод данных.

4) Главное окно: после успешной авторизации открывается главное окно, набор функций которого зависит от вашей роли.

Работа с системой для разных ролей

Для Менеджера:

1) Главное окно (Окно продаж): после входа вы попадаете в главное окно.

Здесь вы можете:

- Просматривать актуальные тарифы.
- Начать оформление новой продажи, нажав кнопку "Новая продажа".
- Начать оформление нового контракта, нажав кнопку "Новый контракт".
- Добавить нового клиента, нажав кнопку "Новый клиент".
- Перейти в редактор данных (доступны только операции просмотра и добавления).

2) Оформление продажи:

- Нажмите кнопку "Новая продажа".
- В открывшемся окне выберите Клиента из выпадающего списка.
- Выберите Товар из выпадающего списка.
- При необходимости добавьте Комментарий.
- Нажмите кнопку "Добавить" для сохранения продажи.

3) Оформление контракта:

- Нажмите кнопку "Новый контракт" (или перейдите через "Редактор данных").
- В открывшемся окне выберите Клиента и Тариф.
- Укажите Статус контракта.
- Нажмите кнопку "Добавить" для сохранения контракта.

4) Добавление клиента:

- Нажмите кнопку "Новый клиент".
- Заполните поля формы (ФИО, Телефон, Email, Адрес).
- Нажмите кнопку "Добавить".

5) Редактор данных: Нажмите кнопку "Редактор данных". Вы можете просматривать данные в таблицах (Клиенты, Товары, Тарифы, Продажи, Контракты), выполнять поиск и сортировку. Добавление новых записей

возможно для некоторых справочников. Кнопки "Изменить" и "Удалить" неактивны.

Для Старшего менеджера:

Функционал аналогичен Менеджеру, но с расширенными правами в "Редакторе данных":

- 1) Редактор данных: Нажмите кнопку "Редактор данных".
 - Вы можете просматривать данные.
 - Вы можете добавлять новые записи в справочники (Клиенты, Товары, Тарифы, Продажи, Контракты).
 - Вы можете изменять существующие записи в справочниках (Клиенты, Контракты), выделив строку и нажав кнопку "Изменить".
 - Кнопка "Удалить" активна для Клиенты и Продажи.

Для Администратора:

Администратор имеет максимальный уровень доступа.

1. Главное окно (Администрация): после входа вы попадаете в окно "Администрация".

- Выберите Салон из выпадающего списка.
- Нажмите кнопку "Анализ", чтобы получить отчет по продажам выбранного салона за текущий и предыдущий месяц.
- Нажмите кнопку "Панель продаж", чтобы перейти к функциям, аналогичным интерфейсу Менеджера.
- Нажмите кнопку "Редактор данных", чтобы получить полный доступ к управлению всеми справочниками.

2. Редактор данных: В "Редакторе данных" Администратор может:

- Просматривать все данные.
- Добавлять новые записи.
- Изменять существующие записи (за исключением Продаж и Авторизации).
- Удалять записи во всех доступных таблицах (за исключением Авторизация).

- Управлять записями в специальных таблицах, таких как Авторизация (управление учетными записями пользователей).

Выход из системы

Для выхода из системы закройте главное окно программы (например, нажав кнопку "Закрыть" в правом верхнем углу окна). Система не требует специальной процедуры выхода.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы была разработана информационная система для автоматизации деятельности отдела продаж салонов сотовой связи. Работа охватывала все ключевые этапы создания программного обеспечения: анализ предметной области, формулирование требований, проектирование архитектуры и базы данных, разработку пользовательского интерфейса и программного кода, а также тестирование и документирование.

Был проведен подробный анализ текущих бизнес-процессов отдела продаж, выявлены недостатки ручной или частично автоматизированной системы учета, и обоснована необходимость внедрения специализированного программного решения. Сформулированы функциональные и нефункциональные требования к системе, определены роли пользователей и их полномочия.

На этапе проектирования были созданы и описаны основные диаграммы: ER-диаграмма, описывающая структуру данных; контекстная диаграмма IDEF0, отражающая систему в целом и её взаимодействие с внешней средой; диаграмма декомпозиции IDEF0, детализирующая внутренние процессы; DFD-диаграмма, показывающая потоки данных; диаграмма Use Case, определяющая функции системы с точки зрения пользователей. Эти диаграммы легли в основу проектирования базы данных и архитектуры приложения.

В рамках реализации была спроектирована и создана реляционная база данных с использованием СУБД Microsoft SQL Server, обеспечивающая целостность и хранение всех необходимых данных. Разработанное программное обеспечение на языке C# с применением технологии Windows Forms предоставляет пользователям удобный интерфейс для выполнения их профессиональных задач: работы с клиентами, оформления продаж и контрактов, управления справочниками. Реализовано разграничение прав доступа в зависимости от роли пользователя (Менеджер, Старший менеджер,

Администратор). Проведено тестирование, подтвердившее корректную работу основных функций системы и её соответствие поставленным требованиям.

Таким образом, в ходе выполнения ВКР была успешно решена задача создания программного продукта, способного автоматизировать ключевые бизнес-процессы отдела продаж. Разработанная информационная система позволяет повысить эффективность работы, снизить вероятность ошибок, упростить анализ данных и тем самым внести вклад в оптимизацию деятельности салона сотовой связи. Работа продемонстрировала способность автора применить теоретические знания по информационным системам и программированию на практике для решения реальной прикладной задачи.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Печатные источники:

1. Алексанян, С.А. Проектирование информационных систем: Учебное пособие / С.А. Алексанян. — 2-е изд., перераб. и доп. — М.: ИД «Форум», 2023. — 256 с. — ISBN 978-5-91131-999-1.
2. Карпов, А.Е. Разработка приложений на C# и .NET: от новичка до профессионала / А.Е. Карпов. — СПб.: Питер, 2024. — 576 с. — ISBN 978-5-4461-2345-6.
3. Кузнецов, В.Д. Системы управления базами данных: теория и практика / В.Д. Кузнецов. — М.: Бином, 2023. — 432 с. — ISBN 978-5-9963-8888-7.
4. Мартин, Р.Ч. Чистая архитектура: Искусство разработки программного обеспечения. — СПб.: Питер, 2022. — 352 с. — ISBN 978-5-496-02308-6.
5. Петров, В.Н. Информационные системы: Учебник для вузов / В.Н. Петров. — 4-е изд., перераб. и доп. — М.: Питер, 2023. — 624 с. — ISBN 978-5-496-02100-6.
6. Пресняков, В.Р. Проектирование баз данных: Учебное пособие / В.Р. Пресняков. — М.: ИНФРА-М, 2024. — 288 с. — ISBN 978-5-16-123456-7.
7. Соммервилл, И. Инженерия программного обеспечения: 10-е издание. — М.: Лори, 2023. — 816 с. — ISBN 978-5-85583-456-7.
8. Фаулер, М. Архитектура корпоративных приложений / М. Фаулер. — М.: Символ-Плюс, 2022. — 544 с. — ISBN 978-5-93286-234-5.

Электронные ресурсы:

1. Microsoft Docs. Документация по C#. URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения: 29.11.2025).
2. Microsoft Docs. Документация по Windows Forms. URL: <https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/?view=netdesktop-6.0> (дата обращения: 29.11.2025).

3. Microsoft Docs. Документация по SQL Server. URL: <https://docs.microsoft.com/ru-ru/sql/sql-server/?view=sql-server-ver16> (дата обращения: 29.11.2025).

4. Хабр. Веб-сайт о разработке. URL: <https://habr.com> (дата обращения: 15.09.2024).

5. Stack Overflow. Веб-сайт вопросов и ответов по программированию. URL: <https://stackoverflow.com> (дата обращения: 20.10.2024).

Справочная литература:

1. ГОСТ Р 51317.4.7-2007/МЭК 61000-4-7:2002. Совместимость технических средств электромагнитная. Устойчивость к наносекундным импульсам. — Введ. 2008-01-01. — М.: Стандартинформ, 2007. — 20 с.

2. Федеральный закон от 27 июля 2006 г. № 149-ФЗ "Об информации, информационных технологиях и о защите информации". // Собрание законодательства РФ. – 2006. – № 31. – Ст. 3448.

3. Федеральный закон от 27 июля 2006 г. № 152-ФЗ "О персональных данных". // Собрание законодательства РФ. – 2006. – № 31. – Ст. 3449.

ПРИЛОЖЕНИЕ

Класс Program.cs:

```
using Program.scr.forms;
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 AuthForm.cs:

```
using Microsoft.Data.SqlClient;
using Program.scr.core;
using Program.scr.core.dbt;

namespace Program.scr.forms
{
    public partial class AuthForm : Form
    {
        public AuthForm()
        {
            InitializeComponent();
        }

        private void button_auth_Click(object sender, EventArgs e)
        {
            string login = textBox_login.Text.Trim();
            string password = textBox_password.Text.Trim();

            if (string.IsNullOrEmpty(login) ||
string.IsNullOrEmpty(password))
            {
                MessageBox.Show("Неверный логин или пароль", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }
        }
}
```

```

try
{
    using (SqlConnection connection = new
SqlConnection(SQL._sqlConnectStr))
    {
        connection.Open();
        using (var query = connection.CreateCommand())
        {
            query.CommandText = "SELECT ID_Employee,
ID_Auth FROM Auth WHERE Login = @login AND Password = @password";
            query.Parameters.AddWithValue("@login",
login);
            query.Parameters.AddWithValue("@password",
password);
            using (var reader = query.ExecuteReader())
            {
                while (reader.Read())
                {
                    if (reader.IsDBNull(0))
                    {
                        MessageBox.Show("Неверный логин
или пароль", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                        return;
                    }

                    Core.ThisEmployee =
DBT_Employee.GetById(reader.GetInt32(0));
                    Core.ThisUser =
DBT_Auth.GetById(reader.GetInt32(1));
                }
            }
        }
    }
    catch (Exception ex) { MessageBox.Show("Неверный логин или
пароль", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);
return; }

    if (Core.ThisUser == null)
    {
        MessageBox.Show("Неверный логин или пароль", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    if (Core.ThisUser.AccessLevel == "Not")
    {
        MessageBox.Show("Нет доступа :(", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
}

```

```

        this.Hide();
        if (Core.ThisUser.AccessLevel == "Admin") new
AdminForm().ShowDialog();
        else if (Core.ThisUser.AccessLevel == "Manager") new
ManagerForm().ShowDialog();
        else if (Core.ThisUser.AccessLevel == "HeadManager") new
ManagerForm().ShowDialog();
        else MessageBox.Show("Роль не опознана!", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Warning);

        Application.Exit();
    }
}
}

```

Класс Analytics:

```

public class Analytics
{
    public static async Task<string> GenerateAnalyticsHtmlAsync(int
selectedSalonId)
    {
        // Пример строки подключения. ЗАМЕНИТЕ НА СВОЮ!
        string connectionString = SQL._sqlConnectStr;

        var htmlBuilder = new StringBuilder();
        htmlBuilder.AppendLine("<html><body>");

        // Текущий и предыдущий месяцы
        DateTime currentDate = DateTime.Today;
        DateTime currentMonthStart = new DateTime(currentDate.Year,
currentDate.Month, 1);
        DateTime currentMonthEnd =
currentMonthStart.AddMonths(1).AddDays(-1);

        DateTime previousMonthStart = currentMonthStart.AddMonths(-1);
        DateTime previousMonthEnd = currentMonthStart.AddDays(-1);

        try
        {
            using (var connection = new
SqlConnection(connectionString))
            {
                await connection.OpenAsync();

                // --- Заголовок ---
                htmlBuilder.AppendLine($"<h2 style='color:
#2E86AB;'>Аналитика для салона №:
{DBT_Salon.GetById(selectedSalonId).Address}</h2>");

```

```

        htmlBuilder.AppendLine($"<p style='color: #5E6C84;
font-style: italic; '>Периоды: {previousMonthStart:MMMM yyyy} vs
{currentMonthStart:MMMM yyyy}</p>");

        // --- 1. Анализ клиентов ---
        htmlBuilder.AppendLine("<h3 style='color:
#F18F01; '>Анализ клиентов</h3>");

        // Запросы для подсчета новых клиентов
        string newClientsQuery = @"
SELECT COUNT(*) FROM Client c
INNER JOIN Sale s ON c.ID_Client = s.ID_Client
INNER JOIN Employee e ON s.ID_Employee = e.ID_Employee
WHERE e.ID_Salon = @SalonId AND s.SaleDate >= @PeriodStart AND
s.SaleDate <= @PeriodEnd
AND NOT EXISTS (
    SELECT 1 FROM Sale prev_s
    INNER JOIN Employee prev_e ON prev_s.ID_Employee =
prev_e.ID_Employee
    WHERE prev_e.ID_Salon = @SalonId AND prev_s.ID_Client =
c.ID_Client
    AND prev_s.SaleDate < @PeriodStart
)";

        int newClientsCurrent = await
GetCountAsync(connection, newClientsQuery, selectedSalonId,
currentMonthStart, currentMonthEnd);
        int newClientsPrevious = await
GetCountAsync(connection, newClientsQuery, selectedSalonId,
previousMonthStart, previousMonthEnd);

        int clientDiff = newClientsCurrent -
newClientsPrevious;
        string clientDiffSign = clientDiff >= 0 ? "+" : "";
        string clientColor = clientDiff >= 0 ? "#009B77" :
"#DD4132";

        htmlBuilder.AppendLine($"<p>Новые клиенты в текущем
месяце: <strong style='color:
#2E86AB; '>{newClientsCurrent}</strong></p>");
        htmlBuilder.AppendLine($"<p>Новые клиенты в предыдущем
месяце: <strong style='color:
#5E6C84; '>{newClientsPrevious}</strong></p>");
        htmlBuilder.AppendLine($"<p>Изменение: <strong
style='color: {clientColor}; '>{clientDiffSign}{clientDiff}</strong> "
+
                (clientDiff > 0 ? "<span
style='color: #009B77; '>(Рост)</span>" : clientDiff < 0 ? "<span
style='color: #DD4132; '>(Снижение)</span>" : "<span style='color:
#5E6C84; '>(Без изменений)</span>") + "</p>");
```

```

        // --- 2. Анализ продаж ---
        htmlBuilder.AppendLine("<h3 style='color: #F18F01;'>Анализ продаж</h3>");

        // Запросы для подсчета продаж и суммы
        string salesQuery = @"
SELECT COUNT(*), ISNULL(SUM(TotalAmount), 0) FROM Sale s
INNER JOIN Employee e ON s.ID_Employee = e.ID_Employee
WHERE e.ID_Salon = @SalonId AND s.SaleDate >= @PeriodStart AND
s.SaleDate <= @PeriodEnd";

        var (salesCurrent, sumCurrent) = await
GetSalesDataAsync(connection, salesQuery, selectedSalonId,
currentMonthStart, currentMonthEnd);
        var (salesPrevious, sumPrevious) = await
GetSalesDataAsync(connection, salesQuery, selectedSalonId,
previousMonthStart, previousMonthEnd);

        int salesDiff = salesCurrent - salesPrevious;
        string salesDiffSign = salesDiff >= 0 ? "+" : "";
        string salesColor = salesDiff >= 0 ? "#009B77" :
"#DD4132";

        decimal sumDiff = sumCurrent - sumPrevious;
        string sumDiffSign = sumDiff >= 0 ? "+" : "";
        string sumColor = sumDiff >= 0 ? "#009B77" :
"#DD4132";

        htmlBuilder.AppendLine($"<p>Продаж в текущем месяце:
<strong style='color: #2E86AB;'>{salesCurrent}</strong></p>");
        htmlBuilder.AppendLine($"<p>Продаж в предыдущем
месяце: <strong style='color:
#5E6C84;'>{salesPrevious}</strong></p>");
        htmlBuilder.AppendLine($"<p>Изменение количества
продаж: <strong style='color:
{salesColor};'>{salesDiffSign}{salesDiff}</strong> " +
(salesDiff > 0 ? "<span
style='color: #009B77;'>(Рост)</span>" : salesDiff < 0 ? "<span
style='color: #DD4132;'>(Снижение)</span>" : "<span style='color:
#5E6C84;'>(Без изменений)</span>") + "</p>");

        htmlBuilder.AppendLine($"<p>Общая сумма продаж в
текущем месяце: <strong style='color:
#2E86AB;'>{sumCurrent:C}</strong></p>");
        htmlBuilder.AppendLine($"<p>Общая сумма продаж в
предыдущем месяце: <strong style='color:
#5E6C84;'>{sumPrevious:C}</strong></p>");
        htmlBuilder.AppendLine($"<p>Изменение общей суммы:
<strong style='color: {sumColor};'>{sumDiffSign}{sumDiff:C}</strong> "
+

```

```

                (sumDiff > 0 ? "<span
style='color: #009B77;'>(Рост)</span>" : sumDiff < 0 ? "<span
style='color: #DD4132;'>(Снижение)</span>" : "<span style='color:
#5E6C84;'>(Без изменений)</span>") + "</p>");

        // --- 3. Средний чек ---
        htmlBuilder.AppendLine("<h3 style='color:
#F18F01;'>Средний чек</h3>");

        decimal avgCurrent = salesCurrent > 0 ? sumCurrent /
salesCurrent : 0;
        decimal avgPrevious = salesPrevious > 0 ? sumPrevious /
salesPrevious : 0;

        decimal avgDiff = avgCurrent - avgPrevious;
        string avgDiffSign = avgDiff >= 0 ? "+" : "";
        string avgColor = avgDiff >= 0 ? "#009B77" :
"#DD4132";

        htmlBuilder.AppendLine($"<p>Средний чек в текущем
месяце: <strong style='color: #2E86AB;'>{avgCurrent:C}</strong></p>");
        htmlBuilder.AppendLine($"<p>Средний чек в предыдущем
месяце: <strong style='color:
#5E6C84;'>{avgPrevious:C}</strong></p>");
        htmlBuilder.AppendLine($"<p>Изменение среднего чека:
<strong style='color: {avgColor};'>{avgDiffSign}{avgDiff:C}</strong> "
+
                (avgDiff > 0 ? "<span
style='color: #009B77;'>(Рост)</span>" : avgDiff < 0 ? "<span
style='color: #DD4132;'>(Снижение)</span>" : "<span style='color:
#5E6C84;'>(Без изменений)</span>") + "</p>");

        // --- 4. Топ-3 проданных продуктов ---
        htmlBuilder.AppendLine("<h3 style='color:
#F18F01;'>Топ-3 проданных продуктов</h3>");

        string topProductsQuery = @"
SELECT TOP 3 p.Name, COUNT(s.ID_Sale) AS SalesCount,
SUM(s.TotalAmount) AS TotalSales
FROM Sale s
INNER JOIN Employee e ON s.ID_Employee = e.ID_Employee
INNER JOIN Product p ON s.ID_Product = p.ID_Product
WHERE e.ID_Salon = @SalonId AND s.SaleDate >= @PeriodStart AND
s.SaleDate <= @PeriodEnd
GROUP BY p.ID_Product, p.Name
ORDER BY SalesCount DESC";

        htmlBuilder.AppendLine("<ul>");
        using (var cmd = new SqlCommand(topProductsQuery,
connection))
        {

```

```

        cmd.Parameters.AddWithValue("@SalonId",
selectedSalonId);
        cmd.Parameters.AddWithValue("@PeriodStart",
currentMonthStart);
        cmd.Parameters.AddWithValue("@PeriodEnd",
currentMonthEnd);

            using (var reader = await
cmd.ExecuteReaderAsync())
{
    while (await reader.ReadAsync())
    {
        string productName = reader.GetString(0);
        int salesCount = reader.GetInt32(1);
        decimal totalSales = reader.GetDecimal(2);

htmlBuilder.AppendLine($"<li><strong>{productName}</strong> - Продаж:
{salesCount}, На сумму: {totalSales:C}</li>");
    }
}
htmlBuilder.AppendLine("</ul>");

// --- 5. Топ-3 активных сотрудников ---
htmlBuilder.AppendLine("<h3 style='color:
#F18F01;'>Топ-3 активных сотрудников (по количеству продаж)</h3>");

        string topEmployeesQuery = @"
SELECT TOP 3 e.FullName, COUNT(s.ID_Sale) AS SalesCount,
SUM(s.TotalAmount) AS TotalSales
FROM Sale s
INNER JOIN Employee e ON s.ID_Employee = e.ID_Employee
WHERE e.ID_Salon = @SalonId AND s.SaleDate >= @PeriodStart AND
s.SaleDate <= @PeriodEnd
GROUP BY e.ID_Employee, e.FullName
ORDER BY SalesCount DESC";

        htmlBuilder.AppendLine("<ul>");
        using (var cmd = new SqlCommand(topEmployeesQuery,
connection))
        {
            cmd.Parameters.AddWithValue("@SalonId",
selectedSalonId);
            cmd.Parameters.AddWithValue("@PeriodStart",
currentMonthStart);
            cmd.Parameters.AddWithValue("@PeriodEnd",
currentMonthEnd);

                using (var reader = await
cmd.ExecuteReaderAsync())
                {

```

```

        while (await reader.ReadAsync())
        {
            string empName = reader.GetString(0);
            int empSalesCount = reader.GetInt32(1);
            decimal empTotalSales =
reader.GetDecimal(2);

htmlBuilder.AppendLine($"<li><strong>{empName}</strong> - Продаж:
{empSalesCount}, На сумму: {empTotalSales:C}</li>");
        }
    }
    htmlBuilder.AppendLine("</ul>");

    // --- Заключение ---
    string conclusionColor = (sumDiff >= 0 && salesDiff >=
0) ? "#009B77" : "#DD4132";
    string conclusionText = (sumDiff >= 0 && salesDiff >=
0) ? "Положительная динамика!" : "Требуется внимание.";
    htmlBuilder.AppendLine($"<p style='color:
{conclusionColor}; font-weight: bold; font-size: 1.1em;'>Вывод:
{conclusionText}</p>");

    } // using connection
}
catch (Exception ex)
{
    // Обработка ошибки
    htmlBuilder.AppendLine($"<p style='color: #DD4132; font-
weight: bold;'>Ошибка при выполнении анализа: {ex.Message}</p>");
}

htmlBuilder.AppendLine("</body></html>");
return htmlBuilder.ToString();
}

// Вспомогательный метод для получения количества
private static async Task<int> GetCountAsync(SqlConnection
connection, string query, int salonId, DateTime start, DateTime end)
{
    using (var cmd = new SqlCommand(query, connection))
    {
        cmd.Parameters.AddWithValue("@SalonId", salonId);
        cmd.Parameters.AddWithValue("@PeriodStart", start);
        cmd.Parameters.AddWithValue("@PeriodEnd", end);
        var result = await cmd.ExecuteScalarAsync();
        return result != null ? Convert.ToInt32(result) : 0;
    }
}

// Вспомогательный метод для получения продаж и суммы

```

```
    private static async Task<(int count, decimal sum)>
GetSalesDataAsync(SqlConnection connection, string query, int salonId,
DateTime start, DateTime end)
{
    using (var cmd = new SqlCommand(query, connection))
    {
        cmd.Parameters.AddWithValue("@SalonId", salonId);
        cmd.Parameters.AddWithValue("@PeriodStart", start);
        cmd.Parameters.AddWithValue("@PeriodEnd", end);
        using (var reader = await cmd.ExecuteReaderAsync())
        {
            if (await reader.ReadAsync())
            {
                int count = reader.GetInt32(0);
                decimal sum = reader.GetDecimal(1);
                return (count, sum);
            }
        }
    }
    return (0, 0);
}
```