

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

Институт компьютерных и инженерных наук  
Кафедра информационных и управляющих систем  
Направление подготовки 09.03.02 – Информационные системы и технологии  
Направленность (профиль) образовательной программы Информационные системы и технологии

**ДОПУСТИТЬ К ЗАЩИТЕ**

Зав. кафедрой

\_\_\_\_\_ А.В. Бушманов

« \_\_\_\_\_ » \_\_\_\_\_ 2025 г.

**БАКАЛАВРСКАЯ РАБОТА**

на тему: Проектирование и разработка веб-приложения обработки заявок  
для ООО «АвтоДВС-28»

Исполнитель

студент группы 1104-об

\_\_\_\_\_

(подпись, дата)

А.В. Руденко

Руководитель

доцент, канд. техн. наук

\_\_\_\_\_

(подпись, дата)

О.В. Жилиндина

Консультант:

по безопасности и  
экологичности

доцент, канд. техн. наук

\_\_\_\_\_

(подпись, дата)

А.Б. Булгаков

Нормоконтроль

инженер кафедры

\_\_\_\_\_

(подпись, дата)

В.Н. Адаменко

Благовещенск, 2025

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

Институт компьютерных и инженерных наук  
Кафедра информационных и управляющих систем  
Направление подготовки 09.03.02 – Информационные системы и технологии  
Направленность (профиль) образовательной программы Информационные системы и технологии

УТВЕРЖДАЮ

Зав. кафедрой

\_\_\_\_\_ А.В. Бушманов

« \_\_\_\_\_ » \_\_\_\_\_ 2025 г.

**ЗАДАНИЕ**

К выпускной квалификационной работе студентки Руденко Анны Витальевны

1. Тема выпускной квалификационной работы: Проектирование и разработка веб-приложения обработки заявок для ООО «АвтоДВС-28»

(утверждена приказом от 14.04.2025 № 980-уч)

2. Срок сдачи студентом законченной работы: 10.06.2025

3. Исходные данные к выпускной квалификационной работе: отчет о прохождении преддипломной практики, нормативная документация, спец. литература

4. Содержание выпускной квалификационной работы: анализ предприятия, проектирование информационной системы, разработка программного обеспечения

5. Перечень материалов приложения: техническое задание, сравнение аналогов

6. Консультанты по выпускной квалификационной работе: консультант по безопасности и экологичности Булгаков А.Б., доцент, канд.техн.наук

7. Дата выдачи задания: 02.10.2024

Руководитель выпускной квалификационной работы: Жилиндина О.В., доцент, канд.техн.наук

Задание принял к исполнению: 02.10.2024: \_\_\_\_\_

## РЕФЕРАТ

Бакалаврская работа содержит 79с., 44 рисунка, 14 таблиц, 2 приложения, 14 источников.

ИНФОРМАЦИОННАЯ СИСТЕМА, SQL, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ, ДОКУМЕНТООБОРОТ, ОРГАНИЗАЦИОННАЯ СТРУКТУРА, CRM, ВЕБ-ПРИЛОЖЕНИЕ

В работе проанализирована деятельность предприятия «АвтоДВС-28», включая организационную структуру, бизнес-процессы и текущую информационную систему.

Исследованы риски информационной безопасности, предложены меры по их минимизации, включая внедрение автоматизированной системы обработки заказов, ролевого доступа и шифрования данных.

Целью работы стало изучение информационной инфраструктуры предприятия, анализ её эффективности и разработка целостного веб-приложения для автоматизации клиентского сервиса и операционной деятельности предприятия.

Основные задачи: оценка текущей информационной системы предприятия, разработка веб-приложения для автоматизации процессов взаимодействия с клиентами, создание интегрированных компонентов системы, проектирование модульной архитектуры платформы, обеспечение безопасности транзакций и целостности данных.

## СОДЕРЖАНИЕ

|                                                           |    |
|-----------------------------------------------------------|----|
| Введение                                                  | 8  |
| 1 Анализ деятельности предприятия «автодвс-28»            | 10 |
| 1.1 Общая характеристика предприятия «АвтоДВС-28»         | 10 |
| 1.1.1 Организационная структура предприятия «АвтоДВС-28»  | 10 |
| 1.1.2 Документооборот предприятия «АвтоДВС-28»            | 11 |
| 1.2 Анализ существующих методов решения задачи            | 14 |
| 1.2.1 WordPress CRM Plugin                                | 14 |
| 1.2.2 Битрикс24                                           | 16 |
| 1.2.3 AmoCRM                                              | 17 |
| 1.3 Формулировка задачи и общей методики ее решения       | 19 |
| 2 Проектирование информационной системы                   | 21 |
| 2.1 Архитектурное проектирование системы                  | 21 |
| 2.2 Выбор технологий разработки                           | 24 |
| 2.3 Проектирование базы данных                            | 24 |
| 2.3.1 Инфологическая модель данных                        | 24 |
| 2.3.2 Обоснование установленных связей                    | 28 |
| 2.3.3 Нормализация отношений                              | 31 |
| 2.3.4 Реляционная модель данных                           | 34 |
| 2.3.5 Логическая модель данных                            | 40 |
| 2.3.6 Физическая модель данных                            | 42 |
| 2.4 Проектирование CRM-модулей                            | 45 |
| 2.4.1 Функциональная структура                            | 45 |
| 2.4.2 Архитектура информационной системы                  | 47 |
| 2.4.3 Проектирование пользовательского интерфейса модулей | 49 |
| 2.5 Проектирование системы безопасности                   | 50 |
| 2.5.1 Аутентификация пользователей                        | 50 |
| 2.5.2 Управление правами доступа                          | 50 |
| 2.6 Проектирование модуля взаимодействия с клиентами      | 51 |

|     |                                                  |    |
|-----|--------------------------------------------------|----|
| 3   | Разработка веб-приложения                        | 53 |
| 3.1 | Реализация клиентской части                      | 53 |
| 3.2 | Разработка серверной логики                      | 54 |
| 3.3 | Реализация базы данных                           | 56 |
| 3.4 | Разработка CRM-панели                            | 57 |
| 3.5 | Разработка страниц для взаимодействия с клиентом | 62 |
| 4   | Безопасность и экологичность                     | 68 |
| 4.1 | Безопасность                                     | 68 |
| 4.2 | Экологичность                                    | 71 |
| 4.3 | Чрезвычайные ситуации                            | 73 |
|     | Заключение                                       | 75 |
|     | Библиографические ссылки                         | 76 |
|     | Библиографический список                         | 78 |
|     | Приложение А                                     | 80 |
|     | Приложение Б                                     | 87 |

## НОРМАТИВНЫЕ ССЫЛКИ

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

СТО СМК 4.2.3.21-2018. Оформление выпускных квалификационных и курсовых работ (проектов);

ГОСТ 19.001-77. Единая система программной документации (ЕСПД). Общие положения;

ГОСТ 19.002-80. ЕСПД. Схемы алгоритмов и программ. Правила выполнения;

ГОСТ 19.003-80. ЕСПД. Схемы алгоритмов и программ. Обозначения условные графические;

ГОСТ 19.102-77. Стадии разработки;

ГОСТ 19.201-78. Техническое задание. Требования к содержанию и оформлению;

ГОСТ 19.404-79. Пояснительная записка. Требования к содержанию и оформлению;

ГОСТ Р 50958-2001. Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности;

ГОСТ Р ИСО 9241-161-2016. Эргономика взаимодействия человек-система. Часть 161. Элементы графического пользовательского интерфейса;

ГОСТ Р 52872-2019. Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Приложение для стационарных и мобильных устройств, иные пользовательские интерфейсы;

ГОСТ Р 50923-96. Дисплей. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения.

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

МФУ – многофункциональное устройство

ООО – общество с ограниченной ответственностью

ОСФР – отделение социального фонда России

ПЭВМ – персональная электронно-вычислительная машина

СУБД – система управления базами данных

ФНС – федеральная налоговая служба

ЧС – чрезвычайная ситуация

ACL – список контроля доступа

CRM – система управления взаимоотношениями с клиентами

DNS – система доменных имён

FK – внешний ключ

НОС – компонент высшего порядка

ISR – инкрементальная статическая регенерация

JWT – веб-токен JSON (стандарт токена аутентификации)

OLAP – оперативная аналитическая обработка данных

PK – первичный ключ

POLP – принцип минимальных привилегий

SDLC – жизненный цикл разработки программного обеспечения

SMTP – простой протокол передачи почты

UML – унифицированный язык моделирования

## ВВЕДЕНИЕ

В условиях современного рынка, где конкуренция становится все более жесткой, наличие эффективной информационной системы для управления продажами и обработкой заказов является критически важным. ООО «Авто-ДВС-28» столкнулось с определёнными трудностями в управлении клиентскими запросами и заказами, что негативно сказывается на эффективности работы и удовлетворенности клиентов.

Основные проблемы, связанные с отсутствием или недостаточной автоматизацией процессов, включают в себя долгое время обработки заказов: заказы клиентов принимались только в формате телефонного звонка и хранились в формате электронной таблицы, что затрудняло поддержание актуальности заказов и среднее время их обработки, сложность поддержания актуальной базы товаров: такая информация, как список товаров, их характеристики и количество штук на складе, хранилась также в отдельном файле в формате электронной таблицы, что требовало дополнительное время на поддержание актуальной базы, высокий процент ошибок при ручной обработке данных: в заполнении электронных таблиц из-за наличия человеческого фактора и отсутствия валидации ввода могли возникать ошибки, что приводило к неожиданным ситуациям и могло испортить впечатления клиентов от работы с предприятием. Разработка сайта продаж и системы CRM позволит ускорить процесс обработки заказов, упростить работу сотрудников по поддержанию актуальной базы товаров. Оптимизированный клиентский сервис, улучшит опыт клиентов, которые будут сотрудничать с предприятием.

Предприятием была поставлена задача разработать и реализовать веб-приложение, состоящее из сайта продаж и информационной системы, которое соответствовало бы требованиям предприятия и было бы экономически эффективным по сравнению с готовыми решениями конкурентов.

Предполагаемое веб-приложение представляет собой решение, которое имеет ряд преимуществ по сравнению с готовыми решениями конкурентов,

таких как индивидуальный подход, в котором были учтены специфические нужды ООО «АвтоДВС-28», что позволяет создать решение, максимально соответствующее его требованиям, имеющее гибкость и масштабируемость, при которых, в отличие от готовых решений, собственная система может быть легко изменена и доработана в соответствии с изменяющимися требованиями бизнеса, при этом плюсом являются низкие затраты на долгосрочное обслуживание – после первоначальной настройки и разработки предприятия не нужно будет платить за лицензии или подписки и все будет находиться под контролем внутренних специалистов, а также стоит упомянуть возможность быстрой адаптации, в рамках которой будет проведено обучение и поддержка на этапе внедрения, что ускорит адаптацию сотрудников к новым принципам обработки заказов и поддержании базы товаров.

Цель данной выпускной квалификационной работы заключается в создании эффективной платформы для взаимодействия клиентов и сотрудников компании. Основное назначение системы заключается в предоставлении клиентам возможности просматривать товары, сгруппированные по категориям и фильтрам, и обеспечения функционала корзины с возможностью создания заявки. Важной составляющей является интегрированная система CRM, предназначенная для обработки заказов и управления данными клиентов. В контексте данной работы веб-приложение будет содержать в себе сайт продаж для создания заявок, CRM для управления процессом обработки заказов и базу данных для хранения информации о сотрудниках, клиентах, товарах и заказах. Разработка веб-приложения для ООО «АвтоДВС-28» представляет собой шаг к увеличению его конкурентоспособности на рынке. Учитывая текущие проблемы, возникновение собственной системы для внутреннего пользования может значительно повысить эффективность обработки заказов и улучшить клиентский сервис.

# 1 АНАЛИЗ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ «АВТОДВС-28»

## 1.1 Общая характеристика предприятия «АвтоДВС-28»

### 1.1.1 Организационная структура предприятия «АвтоДВС-28»

Организационная структура компании, изображенная на рисунке 1, представляет собой систему взаимосвязанных элементов, которые обеспечивают эффективное функционирование бизнеса.

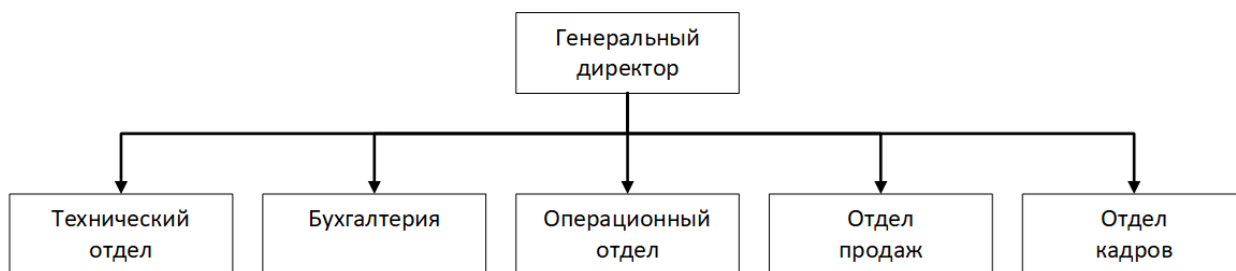


Рисунок 1 – Организационная структура

Организационная структура компании включает шесть ключевых подразделений, каждое из которых выполняет специфические функции, обеспечивающие эффективность деятельности организации.

Центральное место в системе управления занимает генеральный директор, ответственный за формирование стратегии развития компании, определение её долгосрочных целей и координацию работы всех структурных единиц. В его обязанности входит принятие стратегически значимых решений, влияющих на операционную и инвестиционную деятельность, а также представление интересов компании на внешних переговорах и публичных мероприятиях.

Отдел продаж, возглавляемый менеджером, ориентирован на реализацию продуктов и услуг компании. Руководитель подразделения разрабатывает стратегии продвижения, устанавливает целевые показатели для сотрудников и анализирует рыночные тенденции для выявления новых возможностей. Специалисты отдела занимаются привлечением клиентов, проводят презентации продукции, заключают договоры, обеспечивая выполнение плановых показателей по доходам.

Функционирование человеческих ресурсов обеспечивается отделом кадров, который осуществляет подбор и адаптацию персонала, ведет кадровый учет и разрабатывает программы профессионального развития. Важным аспектом деятельности является минимизация текучести кадров и контроль соблюдения трудового законодательства.

Финансовая устойчивость компании обеспечивается бухгалтерией, ответственной за ведение учета операций, подготовку отчетности и взаимодействие с налоговыми органами. Подразделение осуществляет мониторинг бюджетных расходов и оптимизацию финансовых потоков.

Операционный отдел, включающий менеджеров и специалистов по логистике, обеспечивает координацию бизнес-процессов, управление цепочками поставок и обслуживание клиентов. Менеджеры занимаются оптимизацией рабочих процедур, а логисты организуют доставку, управляют запасами и взаимодействуют с транспортными компаниями.

Технический отдел, представленный системным администратором, поддерживает функционирование информационной инфраструктуры. Его задачи охватывают обслуживание сетевых ресурсов, обеспечение кибербезопасности, обновление программного обеспечения и техническую поддержку сотрудников. Совокупность указанных подразделений формирует целостную систему, направленную на достижение стратегических и операционных целей компании.

#### 1.1.2 Документооборот предприятия «АвтоДВС-28»

Внешний документооборот охватывает взаимодействие компании с контрагентами, государственными органами и партнёрами, рисунок 2.

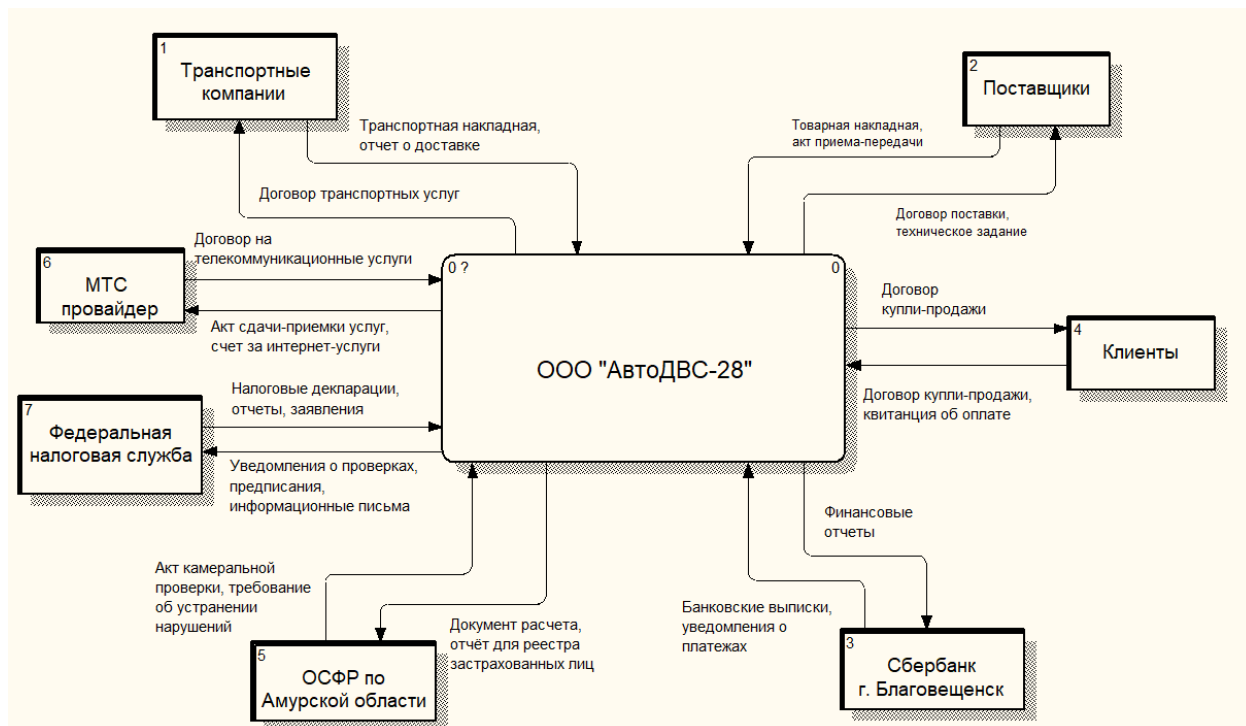


Рисунок 2 – Внешний документооборот

Клиенты инициируют процессы, направляя заказы и платежные поручения, на что «АвтоДВС-28» отвечает заключением договоров купли-продажи и актами сверки взаиморасчётов. Поставщики получают от операционного отдела спецификации заказов и протоколы разногласий, а взамен предоставляют товарные накладные и акты приёма-передачи. Транспортные компании взаимодействуют с операционным отделом через договоры на оказание услуг и графики поставок, подтверждая доставку транспортными накладными. Финансовые учреждения, такие как Сбербанк города, обмениваются с бухгалтерией выписками по счетам и уведомлениями о платежах, а государственные структуры, включая ОСФР и ФНС, направляют требования о проверках, на которые компания реагирует расчётами страховых взносов и отчётами для реестров, регулярно направляет в ФНС налоговые декларации, финансовые отчёты и заявления, обеспечивая соблюдение фискальных обязательств.

Внутренний документооборот компании «АвтоДВС-28» представляет собой систему взаимодействия между структурными подразделениями, обеспечивающую управление бизнес-процессами, рисунок 3.

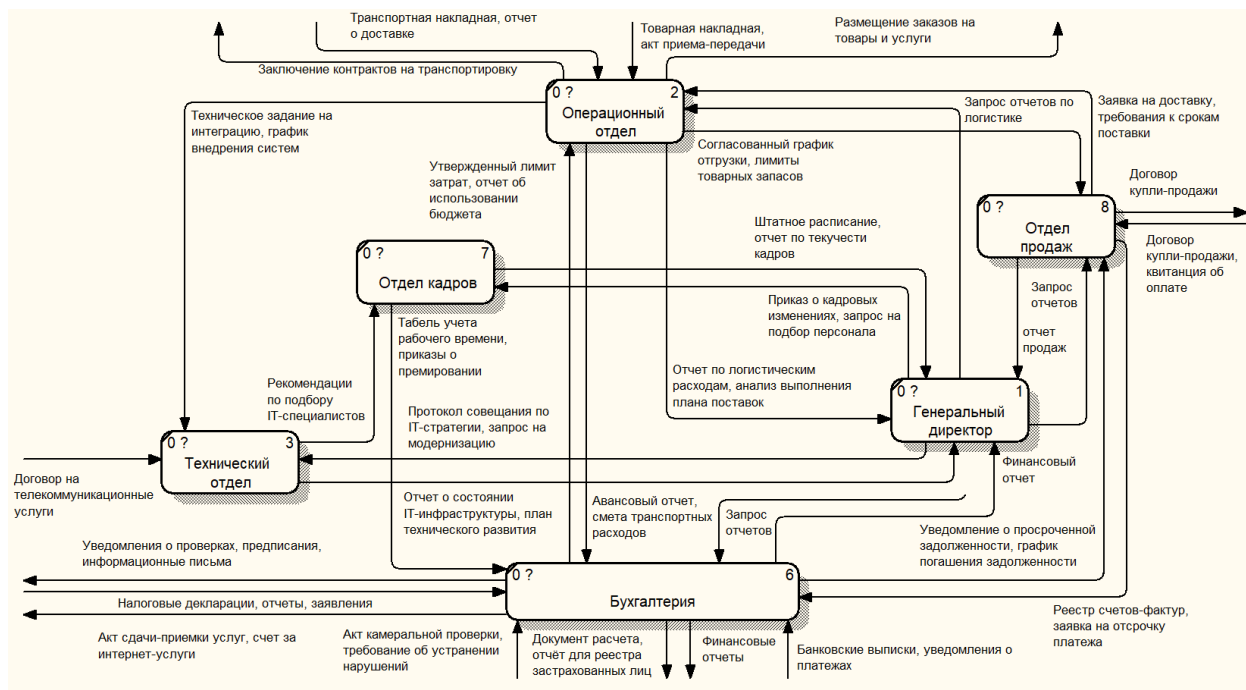


Рисунок 3 – Внутренний документооборот

Основными участниками являются технический отдел, отдел кадров, операционный отдел, отдел продаж, бухгалтерия и руководство в лице генерального директора. Каждое подразделение генерирует документы, которые циркулируют внутри организации, согласуя задачи и контролируя выполнение решений. Например, технический отдел взаимодействует с отделом кадров, предоставляя требования к квалификации ИТ-специалистов, и с генеральным директором, формируя отчёты о состоянии ИТ-инфраструктуры. Операционный отдел координирует логистику, обмениваясь с бухгалтерией авансовыми отчётами и сметами расходов, а отдел продаж передаёт в бухгалтерию реестры счетов-фактур для учёта финансовых операций. Генеральный директор выступает центральным звеном, запрашивая отчёты у всех подразделений и утверждая стратегические решения.

Взаимосвязь внешнего и внутреннего документооборота проявляется в их взаимном влиянии на бизнес-процессы. Например, договор купли-продажи, сформированный отделом продаж для клиента (внешний документ), становится основанием для операционного отдела организовать доставку, что влечёт

заключение договора с транспортной компанией. Полученные от поставщиков товарные накладные (внешний поток) поступают в операционный отдел, который передаёт их в бухгалтерию для проведения платежей, формируя внутренний финансовый отчёт. Акт камеральной проверки от ОСФР или ФНС инициирует внутренний процесс подготовки ответных документов бухгалтерией, интегрируя требования регуляторов во внутренние процедуры компании. Таким образом, внешние документы запускают внутренние бизнес-процессы, а результаты внутренней работы оформляются во внешние юридически значимые акты, обеспечивая непрерывность и прозрачность бизнес-операций.

## **1.2 Анализ существующих методов решения задачи**

В современном бизнесе автоматизация процессов обработки заявок и управления взаимоотношениями с клиентами (CRM) стала неотъемлемой частью эффективного функционирования компаний. В данном разделе рассмотрим существующие решения, которые применяются для достижения этих целей, а именно: плагин CRM для WordPress, набор CRM инструментов Битрикс24 и платформу AmoCRM. В приложении Б приведена таблица, в которой сравниваются основные показатели аналогичных решений.

### **1.2.1 WordPress CRM Plugin**

Плагин CRM для WordPress представляет собой расширение функционала популярных систем управления содержимым (CMS) на базе WordPress. Данный инструмент позволяет пользователям управлять отношениями с клиентами непосредственно на платформе WordPress, рисунок 4.

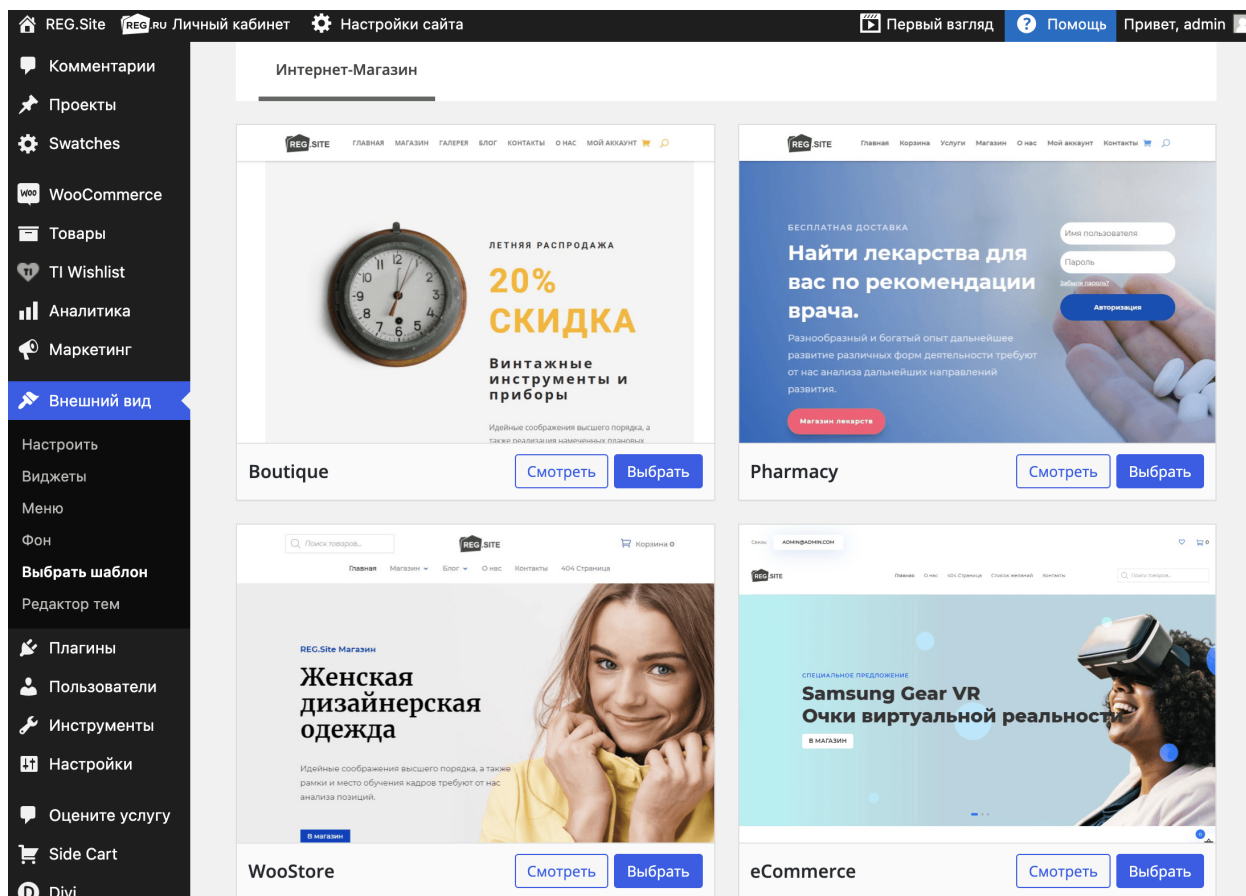


Рисунок 4 – Пример интерфейса с использованием WordPress CRM

К основным преимуществам относится способность данных решений интегрироваться с действующими веб-сайтами, что исключает необходимость разработки отдельного приложения и ускоряет внедрение инструментов управления клиентскими отношениями. Гибкость настройки позволяет адаптировать функционал под конкретные бизнес-процессы за счёт добавления пользовательских полей и модулей. Кроме того, экономическая доступность выражается в наличии разнообразных тарифных моделей, включая бесплатные версии, что снижает финансовую нагрузку на малые предприятия.

К числу недостатков относится ограниченность функционала: многие плагины не поддерживают расширенные возможности, необходимые для средних и крупных компаний, такие как глубокая аналитика или автоматизация сложных рабочих процессов. Ещё одним ограничением является зависимость от платформы WordPress: стабильность работы CRM-системы напрямую

связана с обновлениями и технической поддержкой CMS, что может приводить к рискам прерывания процессов при изменениях архитектуры или возникновении конфликтов совместимости.

### 1.2.2 Битрикс24

Битрикс24 — это многофункциональная и комплексная платформа для всестороннего управления бизнесом, которая включает в себя разнообразные инструменты для управления взаимоотношениями с клиентами (CRM), а также инструменты, способствующие совместной работе сотрудников, управления задачами и проектами. Эта платформа предоставляет сотрудникам широкий набор бизнес-приложений, которые помогают автоматизировать ключевые процессы и повышать общую эффективность работы компании. Битрикс24 направлен на удовлетворение потребностей как малых, так и крупных организаций, обеспечивая гибкость и возможности для масштабирования функционала в зависимости от роста бизнеса, рисунок 5.

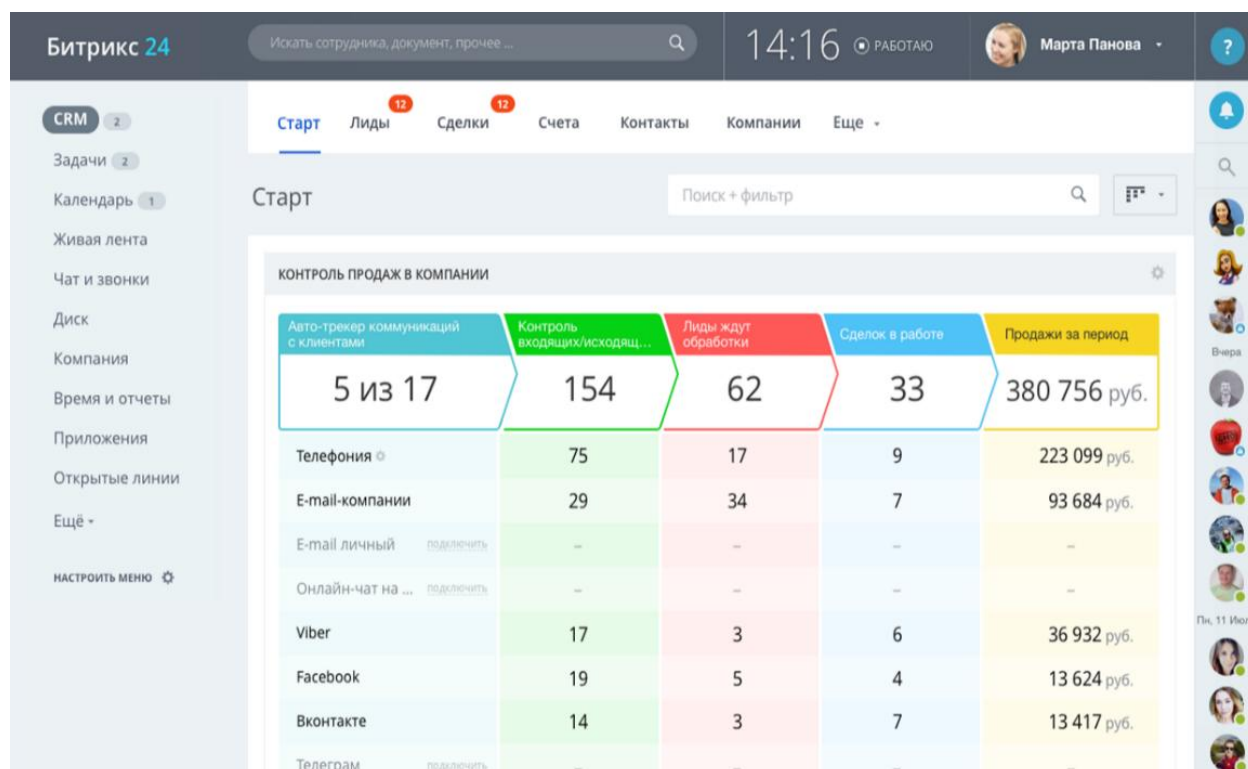


Рисунок 5 – Пример интерфейса Битрикс24

Важным преимуществом является доступность: решение поддерживает как облачное развертывание, так и локальную установку, что обеспечивает гибкость в зависимости от требований бизнеса. Дополнительным аспектом выступает поддержка мобильных приложений, позволяющих сотрудникам взаимодействовать с системой удаленно, что повышает оперативность работы.

К недостаткам платформы относится сложность первоначальной настройки, обусловленная широким спектром функций. Это может потребовать дополнительного времени на обучение сотрудников и адаптацию рабочих процессов. Ещё одним ограничением является стоимость: несмотря на наличие бесплатного тарифного плана, доступ к расширенным функциям, таким как углубленная аналитика или специализированные модули, предусмотрен только в платных подписках, которые могут быть экономически затратными для малых предприятий.

### 1.2.3 AmoCRM

AmoCRM – это облачное программное-обеспечение-как-услуга (SaaS), которое уделяет особое внимание автоматизации процессов продаж и эффективному управлению взаимоотношениями с клиентами. Данная платформа предоставляет пользователям интуитивно понятный и дружелюбный интерфейс, который позволяет легко ориентироваться в функционале и быстро осваивать его. Кроме того, AmoCRM предлагает гибкие возможности для работы с клиентскими данными, что позволяет адаптировать систему под уникальные бизнес-процессы и требования. Благодаря широким функциональным возможностям и удобному дизайну, пользователи могут оптимизировать своё время и усилия, сосредоточившись на продажах и взаимодействии с клиентами, что в свою очередь способствует увеличению доходности бизнеса и повышению уровня обслуживания клиентов, рисунок 6.

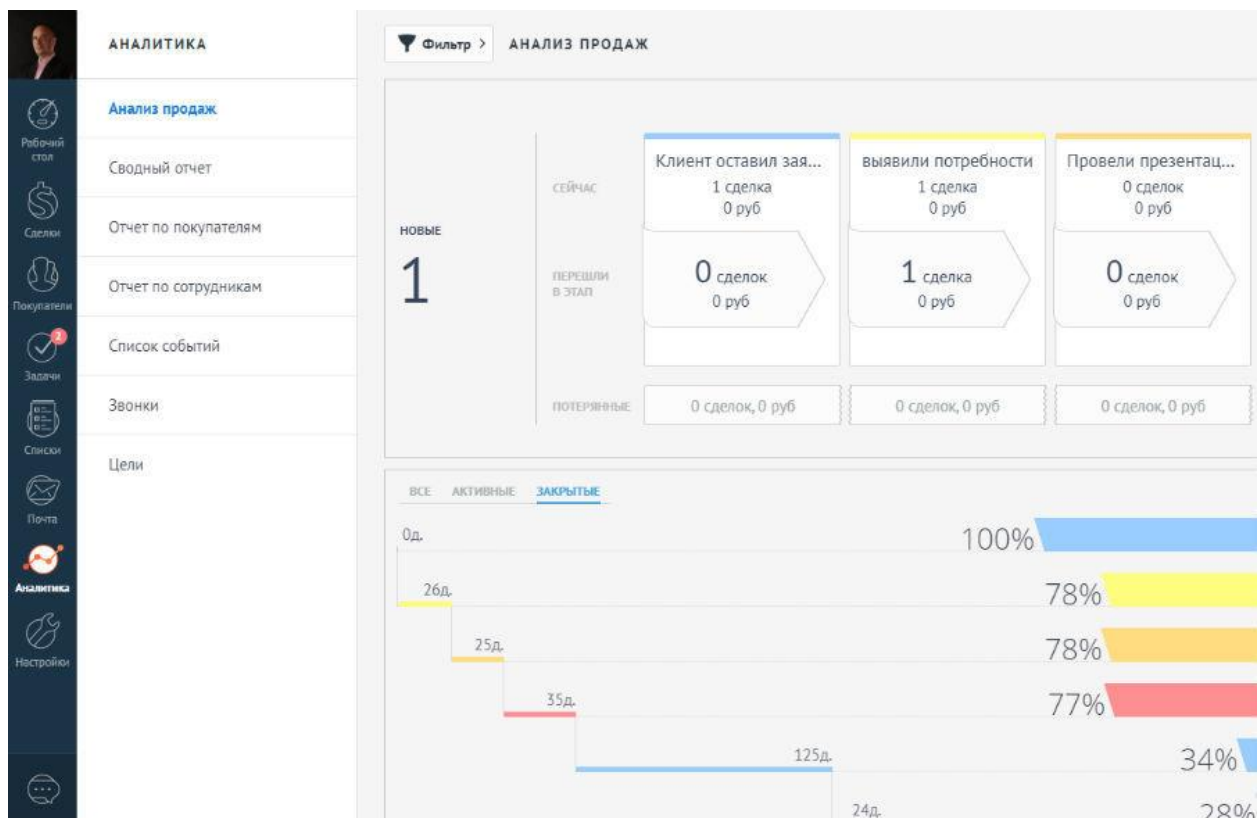


Рисунок 6 – Пример интерфейса AmoCRM

AmoCRM характеризуется рядом преимуществ, делающих её популярным решением для управления клиентскими отношениями. Ключевым достоинством является интуитивно понятный интерфейс, обеспечивающий комфортную работу сотрудников вне зависимости от их технической подготовки. Важным аспектом выступает возможность автоматизации рутинных задач, таких как обработка заявок, рассылка уведомлений или формирование отчетов, что способствует оптимизации временных ресурсов. Платформа также поддерживает интеграцию с широким спектром сторонних сервисов, включая почтовые клиенты, мессенджеры и инструменты аналитики, что расширяет функциональные возможности взаимодействия с клиентами.

Среди недостатков AmoCRM отмечается ограниченная глубина кастомизации: в сравнении с такими решениями, как Битрикс24, платформа предоставляет меньше возможностей для адаптации под узкоспециализированные бизнес-процессы. Кроме того, настройка сложных автоматизированных сценариев, требующих многоуровневой логики, может вызывать затруднения у

пользователей, что увеличивает временные затраты на обучение и внедрение системы. Эти факторы следует учитывать при выборе платформы для компаний с нестандартными операционными требованиями.

### **1.3 Формулировка задачи и общей методики ее решения**

Основной целью данной работы является разработка и внедрение веб-приложения, обеспечивающего эффективное взаимодействие между клиентами и сотрудниками компании посредством веб-приложения и интегрированной CRM-платформы, которое будет достаточно гибким и экономически эффективным в сравнении с аналогичными решениями. Для достижения поставленной задачи предусмотрено выполнение ряда ключевых этапов.

Первостепенным направлением выступает анализ требований сотрудников, включающий документирование функциональных потребностей клиентов и сотрудников в части веб-интерфейса и системы управления заказами. На этапе проектирования архитектуры системы разрабатывается детальная структура приложения с акцентом на разделение пользовательского интерфейса и CRM-модуля для сотрудников. Важным компонентом работы является создание реляционной базы данных, обеспечивающей надёжное хранение и обработку данных о товарах, клиентах и заказах с соблюдением требований к целостности и доступности информации. Заключительный этап предполагает разработку высокопроизводительного и безопасного веб-приложения с использованием фреймворка Next.js, ориентированного на обеспечение удобства взаимодействия для конечных пользователей. Методологическая основа исследования базируется на применении стандартного жизненного цикла разработки программного обеспечения (SDLC), структурированного в три ключевых этапа. На первом этапе, посвящённом сбору и анализу требований, проводятся интервью с сотрудниками компании для выявления потребностей в функционале системы, анализируются существующие решения с целью определения их преимуществ и недостатков, а также формируется техническое задание. Особое внимание уделяется анализу продуктовой линейки и

нормализации структуры базы данных для проектирования соответствующих моделей данных.

Этап проектирования системы включает создание UML-диаграмм (классов и последовательностей) для визуализации архитектурных компонентов и их взаимодействий, а также разработку схемы данных в СУБД с детализацией взаимосвязей между таблицами.

На этапе разработки и реализации формируется файловая структура проекта с использованием системы контроля версий Git, после чего осуществляется программирование веб-приложения на JavaScript с применением фреймворка Next.js и библиотеки React для построения пользовательского интерфейса. Взаимодействие с базой данных SQL реализуется через ORM-библиотеку Prisma, что упрощает выполнение SQL-запросов и обеспечение целостности данных. Данный подход обеспечивает поэтапное выполнение задач с акцентом на соответствие функциональным требованиям и техническую надежность системы.

## 2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

### 2.1 Архитектурное проектирование системы

В процессе проектирования информационной системы одной из основных задач является создание логической структуры, которая позволяет четко выявить взаимосвязи между компонентами системы и их функциональные возможности. Для достижения этой цели было принято решение использовать унифицированный язык моделирования (UML). Этот язык представляет собой общепринятый стандарт для визуализации, спецификации, конструирования и документирования программных систем.

Диаграмма вариантов использования – это один из видов диаграмм в нотации UML (Unified Modeling Language), которая широко используется в области системного анализа и проектирования программного обеспечения. UML предоставляет стандартизированный способ визуализации, спецификации, конструирования и документирования программных систем, и диаграммы вариантов использования играют ключевую роль в этом процессе, представлена на рисунке 7. Она служит для визуального представления взаимодействий между пользователями системы и самими функциональными возможностями, которые эта система предоставляет. Основная цель диаграммы заключается в том, чтобы четко обозначить, какие именно действия могут выполнять пользователи, а также какие требования предъявляются к системе в процессе выполнения этих действий.

В диаграмме вариантов использования изображаются акторы, которые могут представлять как пользователей, так и другие системы, взаимодействующие с целевой системой. Каждый актор обозначает конкретную роль, которую он выполняет в процессе взаимодействия. Варианты использования, в свою очередь, представляют собой функции или услуги, которые система предоставляет акторам. Эти варианты использования описывают сценарии, в которых акторы осуществляют свои действия и достигают определенных целей.

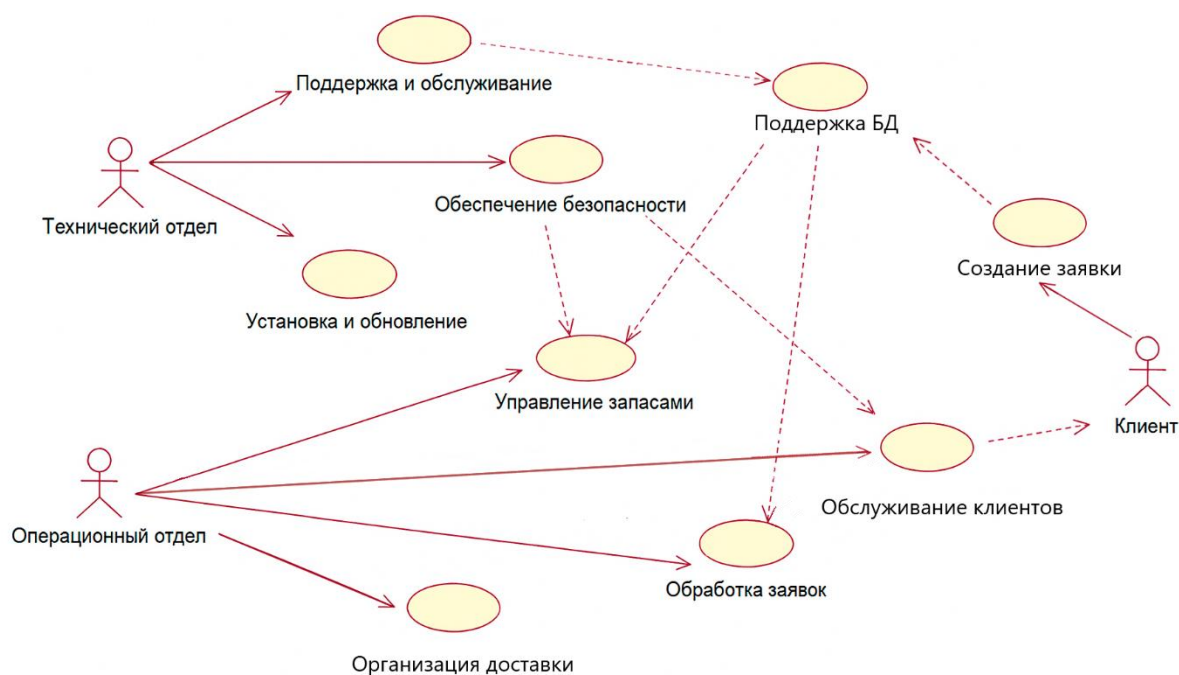


Рисунок 7 – Диаграмма прецедентов

Диаграмма вариантов использования является эффективным средством для выявления и формулирования требований к системе на ранних этапах разработки. Визуализация взаимодействий способствует лучшему пониманию процессов и упрощает коммуникацию между участниками проекта.

Для моделирования взаимодействий между объектами в рамках системы используется диаграмма последовательности, рисунок 8. Она представляет собой один из ключевых типов диаграмм в нотации UML (Unified Modeling Language). Основное предназначение диаграммы последовательности заключается в визуализации динамических аспектов системы, а именно в отображении порядка выполнения операций и обмена сообщениями между различными компонентами.

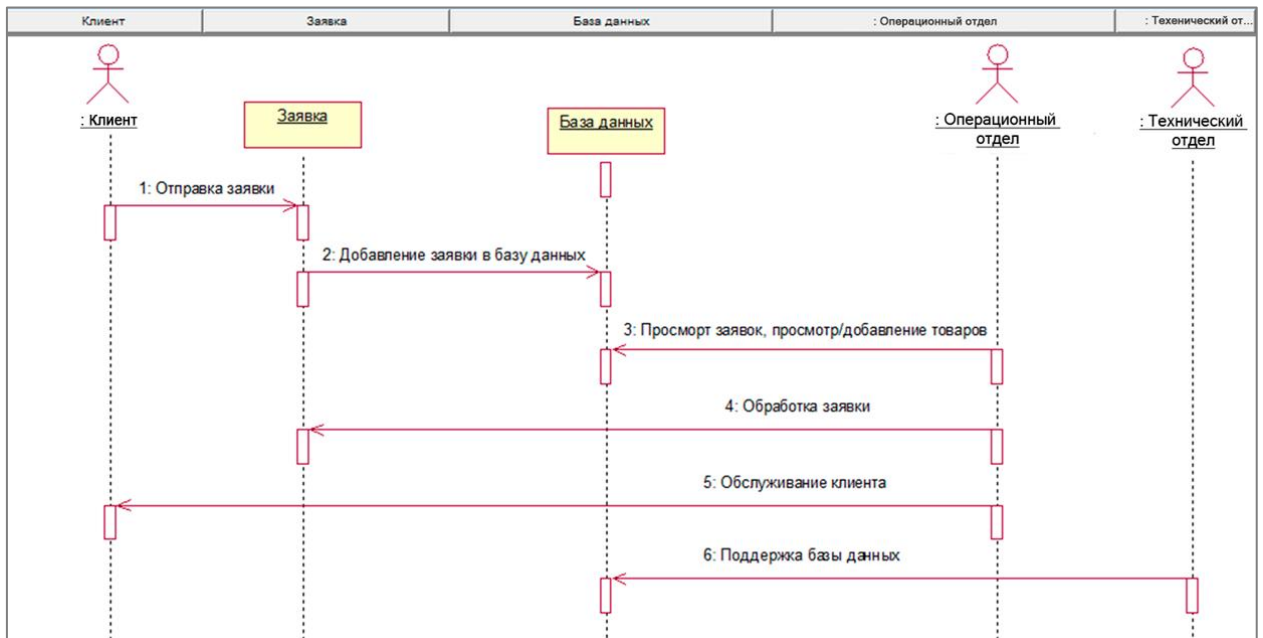


Рисунок 8 – Диаграмма последовательности

В данной диаграмме акцентируется внимание на временной последовательности взаимодействий, что позволяет проиллюстрировать, как объекты взаимодействуют друг с другом во времени. Каждый объект представлен в виде вертикальной линии, называемой «жизненной линией», которая демонстрирует активность объекта на протяжении всего взаимодействия. Сообщения, передаваемые между объектами, отображаются стрелками, направленными от отправителя к получателю, что позволяет наглядно представить последовательность вызовов методов и обмена данными.

Диаграмма последовательности является важным инструментом для анализа и проектирования программных систем, так как она способствует более глубокому пониманию логики работы системы и взаимодействия её компонентов. Благодаря своей наглядности и простоте восприятия, диаграмма последовательности служит эффективным средством коммуникации между членами команды разработки и другими заинтересованными сторонами, обеспечивая ясность и согласованность в понимании процессов, происходящих в системе.

## **2.2 Выбор технологий разработки**

При выборе технологического стека для реализации веб-приложения ключевым критерием стало соответствие инструментов масштабируемости проекта и требованиям к производительности. Фреймворк Next.js был выбран в качестве базовой платформы разработки благодаря его интеграции с библиотекой React, поддержке серверного рендеринга и наличию встроенных механизмов маршрутизации [1]. Важным фактором стала экосистема фреймворка, предоставляющая готовые решения для оптимизации изображений, международной локализации и кэширования данных, что существенно сокращает время разработки.

Для организации хранения данных рассматривались реляционные СУБД MySQL и SQLite. MySQL была выбрана для производственного окружения благодаря возможности горизонтального масштабирования и надежным механизмам репликации данных. В тестовой среде предполагается использование SQLite, что обусловлено простотой настройки и отсутствием необходимости в отдельном сервере баз данных. Обе системы обеспечивают совместимость с ORM-инструментарием, позволяющим абстрагировать уровень работы с данными от конкретной реализации СУБД.

## **2.3 Проектирование базы данных**

### **2.3.1 Инфологическая модель данных**

Инфологическое проектирование в базах данных является одним из этапов процесса разработки базы данных. Оно направлено на создание структуры данных, которая будет использоваться для хранения, организации и управления информацией в базе данных.

Целью данного процесса является обеспечение эффективности и надежности работы базы данных. При инфологическом проектировании необходимо определить и описать сущности, которые будут представляться в базе данных.

Сущность – это представление некоторого объекта или понятия в информационной системе. Она содержит набор атрибутов, которые описывают

характеристики этого объекта. В процессе инфологического проектирования необходимо определить все сущности, их атрибуты и связи между ними.

Для начала проектирования проводится анализ предметной области, чтобы полностью понять суть информационной системы и ее задачи. Также важно учесть потребности пользователей и требования заказчика к проекту.

Каждая сущность имеет свои атрибуты – это конкретные данные, которые описывают данную сущность. Атрибуты могут быть простыми (например, имя или номер телефона) или составными (например, адрес, который включает в себя город, улицу и номер дома).

Инфологическое проектирование позволяет создать четкую и структурированную базу данных. Оно помогает оптимизировать процессы в информационной системе и повысить ее функциональность. Кроме того, правильное определение сущностей и атрибутов упрощает разработку и сопровождение системы в дальнейшем.

Концептуальная модель представляет собой абстрактное представление, которое позволяет описать сущности и связи между ними.

В процессе создания концептуальной модели сущности и атрибуты могут быть представлены в виде диаграмм, таблиц или других графических средств, которые помогают визуализировать взаимосвязи и зависимости. Каждая сущность и ее атрибуты могут быть представлены с использованием специальных символов и обозначений, что делает модель более наглядной и понятной.

Таким образом, изучив предметную область и техническое задание, внесем данные в таблицы 1–6. Ключевой атрибут в каждой сущности помечен буквами РК.

Первая сущность «Сотрудник» включает данные о работниках отдела, где для каждого сотрудника фиксируются должность и пароль.

Таблица 1 – Сущность «Сотрудник»

| Название атрибута   | Описание атрибута | Тип данных | Диапазон значений | Пример атрибута       |
|---------------------|-------------------|------------|-------------------|-----------------------|
| Код сотрудника (РК) |                   | Числовой   | >0                | 1                     |
| Электронная почта   |                   | Текст      | —                 | example@email.ru      |
| Хеш пароля          |                   | Текст      | —                 | \$2a\$12\$...XPD53LtC |
| Роль                |                   | Текст      | —                 | ADMIN                 |
| Статус              |                   | Логический | —                 | false                 |
| Имя                 |                   | Текст      | —                 | Иван                  |
| Фамилия             |                   | Текст      | —                 | Иванов                |
| Телефон             |                   | Текст      | —                 | +79001234567          |

Вторая сущность «Клиент» предназначена для хранения контактной информации о клиентах.

Таблица 2 – Сущность «Клиент»

| Название атрибута | Описание атрибута | Тип данных | Диапазон значений | Пример атрибута  |
|-------------------|-------------------|------------|-------------------|------------------|
| Код клиента (РК)  |                   | Числовой   | >0                | 1                |
| Электронная почта |                   | Текст      | —                 | example@email.ru |
| Имя               |                   | Текст      | —                 | Иван             |
| Фамилия           |                   | Текст      | —                 | Иванов           |
| Телефон           |                   | Текст      | —                 | +79001234567     |

Сущность «Заявка» содержит в себе информацию о покупателе, товарах и текущем статусе обработки.

Таблица 3 – Сущность «Заявка»

| Название атрибута | Описание атрибута | Тип данных | Диапазон значений | Пример атрибута |
|-------------------|-------------------|------------|-------------------|-----------------|
| Номер заявки (PK) |                   | Числовой   | >0                | 1               |
| Статус            |                   | Текст      | –                 | COMPLETED       |

Сущность «Двигатель» содержит в себе информацию о наименовании двигателя, к которому могут относиться некоторые товары.

Таблица 4 – Сущность «Двигатель»

| Название атрибута  | Описание атрибута | Тип данных | Диапазон значений | Пример атрибута |
|--------------------|-------------------|------------|-------------------|-----------------|
| Код двигателя (PK) |                   | Числовой   | >0                | 1               |
| Наименование       |                   | Текст      | –                 | ISBe            |

Сущность «Категория» содержит в себе информацию о категории товаров, а также уникальный идентификатор категории, который содержится в URL-адресе этой категории.

Таблица 5 – Сущность «Категория»

| Название атрибута  | Описание атрибута | Тип данных | Диапазон значений | Пример атрибута |
|--------------------|-------------------|------------|-------------------|-----------------|
| Код категории (PK) |                   | Числовой   | >0                | 1               |
| Наименование       |                   | Текст      | –                 | Насосы          |
| URL-адрес          |                   | Текст      | –                 | nasos           |

Сущность «Товар» содержит в себе основную информацию о товаре, например, артикул, наименование, описание, цену, количество на складе.

Таблица 6 – Сущность «Товар»

| Название атрибута    | Описание атрибута | Тип данных | Диапазон значений | Пример атрибута        |
|----------------------|-------------------|------------|-------------------|------------------------|
| Код товара (РК)      |                   | Числовой   | >0                | 1                      |
| Артикул              |                   | Текст      | –                 | 4891116                |
| Наименование         |                   | Текст      | –                 | Насос ГУРа             |
| Описание             |                   | Текст      | –                 | Материал/Вид техники   |
| Цена                 |                   | Числовой   | >0                | 10000                  |
| Количество на складе |                   | Числовой   | ≥0                | 120                    |
| Изображение          |                   | Текст      | –                 | /www/images/tovar1.jpg |

### 2.3.2 Обоснование установленных связей

Модельные представления, основанные на анализе семантики данных, иногда называют семантическими моделями. Одним из распространенных средств спецификации модельных представлений этого типа является т.н. «модель сущность – связь» (Entity – Relationship Model). Модель «сущность – связь» (ER-модель) была предложена П. Ченом в 1976 году как средство «ручного» проектирования баз данных. Модель «сущность – связь» представляет собой набор концепций, используемых для описания логической структуры базы данных.

Модель «сущность – связь» основана на диаграммной технике. Для представления различных аспектов структуры данных (объектов, свойств объектов, связей между объектами, свойств связей и других) используются графические средства. За многолетнюю историю использования модель «сущность – связь» претерпела различные модификации, известны несколько ее нотаций. Наиболее популярной в настоящее время является нотация, в большей степени приближенная к реальному процессу проектирования баз данных, т. н. методология информационного моделирования IDEF1X.

Самой известной графической нотацией или же диаграммой является нотация Чена. Множества сущностей изображаются в виде прямоугольников, множества отношений изображаются в виде ромбов. Если сущность участвует в отношении, они связаны линией. Если отношение не является обязательным, то линия пунктирная. Атрибуты изображаются в виде овалов и связываются линией с одним отношением или с одной сущностью.

Между типами сущностей различают следующих 3 типа связей:

- «один-к-одному» или 1:1. Это значит, что одному экземпляру некоторой сущности может соответствовать только один экземпляр другой сущности;
- «один-ко-многим» или 1:M. Это значит, что одному экземпляру сущности может соответствовать любое количество (M) экземпляров другой сущности. Если известно значение максимального количества экземпляров, то это значение указывается вместо символа M;
- «много-к-многим» или M:N. Это означает, что нескольким экземплярам одной сущности может соответствовать несколько экземпляров другой сущности.

На рисунке 9 представлена связь «один-ко-многим» между сущностями «Клиент – Заявка». Одним клиентом может быть создано много заявок, при этом заявка принадлежит только одному клиенту.



Рисунок 9 – Связь «Клиент – Заявка»

На рисунке 10 представлена связь «один-ко-многим» между сущностями «Сотрудник – Заявка». Одним сотрудником может обрабатываться множество заявок, при этом множество заявок может выполняться одним сотрудником.

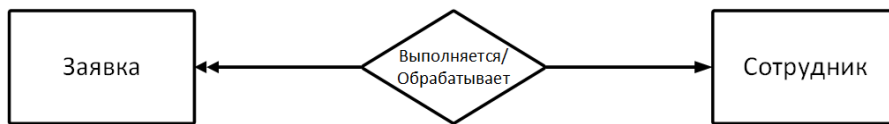


Рисунок 10 – Связь «Сотрудник – Заявка»

На рисунке 11 представлена связь «один-ко-многим» между сущностями «Товар – Двигатель». Один товар может быть привязан к одному типу двигателей, при этом к одному двигателю может быть привязано множество товаров.



Рисунок 11 – Связь «Товар – Двигатель»

На рисунке 12 представлена связь «многие-ко-многим» между сущностями «Товар – Категория». Множество товаров может быть привязано ко множеству категорий, при этом каждая категория может содержать в себе множество товаров.



Рисунок 12 – Связь «Товар – Категория»

На рисунке 13 представлена связь «многие-ко-многим» между сущностями «Товар – Заявка». Множество товаров может быть привязано ко множеству заявок, при этом множество заявок может содержать в себе по несколько товаров.



Рисунок 13 – Связь «Товар – Заявка»

Таким образом, получившиеся связи охватывают вне необходимые процессы, которые будут происходить в системе.

### 2.3.3 Нормализация отношений

Нормализация отношений в базе данных является важным процессом, направленным на улучшение структуры и эффективности хранения данных. Этот метод позволяет снизить дублирование информации, упростить систему запросов к базе данных и обеспечить целостность данных.

Одним из ключевых принципов нормализации является разделение информации на отдельные таблицы, чтобы данные хранились в структурированном виде. Это позволяет избежать избыточности информации, что повышает надежность базы данных и облегчает ее поддержку и обновление.

Существуют несколько уровней нормализации, включая 1-ю нормальную форму (1НФ), 2-ю нормальную форму (2НФ) и 3-ю нормальную форму (3НФ). Первая нормальная форма (1НФ) требует, чтобы каждый столбец в таблице имел однозначное значение и не допускал повторений. Это обеспечивает атомарность данных и предотвращает «множественные значения». Кроме того, для достижения 1НФ необходимо определить первичный ключ каждой таблицы. Для приведения данных к 1НФ необходимо выполнить следующие шаги: определить сущности и их атрибуты, создать отдельные таблицы для каждой сущности, добавить уникальный идентификатор к каждой таблице, устранить повторяющиеся группы атрибутов, устранить многозначные атрибуты.

В каждой сущности определён первичный ключ, отсутствуют сложные атрибуты, каждый из них однозначен, соответственно, они приведены к 1НФ (рисунок 14).



Рисунок 14 – Итоговые сущности в 1НФ

Вторая нормальная форма (2НФ) требует, чтобы каждый столбец в таблице зависел от полного состава первичного ключа, а не от отдельной его части. Это позволяет устранить зависимости между данными, что повышает эффективность запросов и обновлений базы данных.

2НФ позволяет избежать аномалий обновления данных, которые могут возникнуть при изменении или удалении записей. Он также устраняет избыточность данных, связанную с повторением атрибутов в отношении. Нормализация до 2НФ обеспечивает более эффективное хранение данных и более гибкую структуру для обработки запросов. Для приведения таблиц ко второй нормальной форме (2НФ) нужно выполнить следующие действия: все таблицы должны находиться в первой нормальной форме (1НФ), должно быть соблюдено условие – любой столбец, который не является ключом (в том числе внешним), должен зависеть от первичного ключа, если данные, содержащиеся в столбце, не имеют отношения к ключу, который описывает строку, то их следует отделять в свою отдельную таблицу (в старую таблицу надо возвращать первичный ключ). Рассматриваемые отношения уже приведены к 2НФ, рисунок 15.

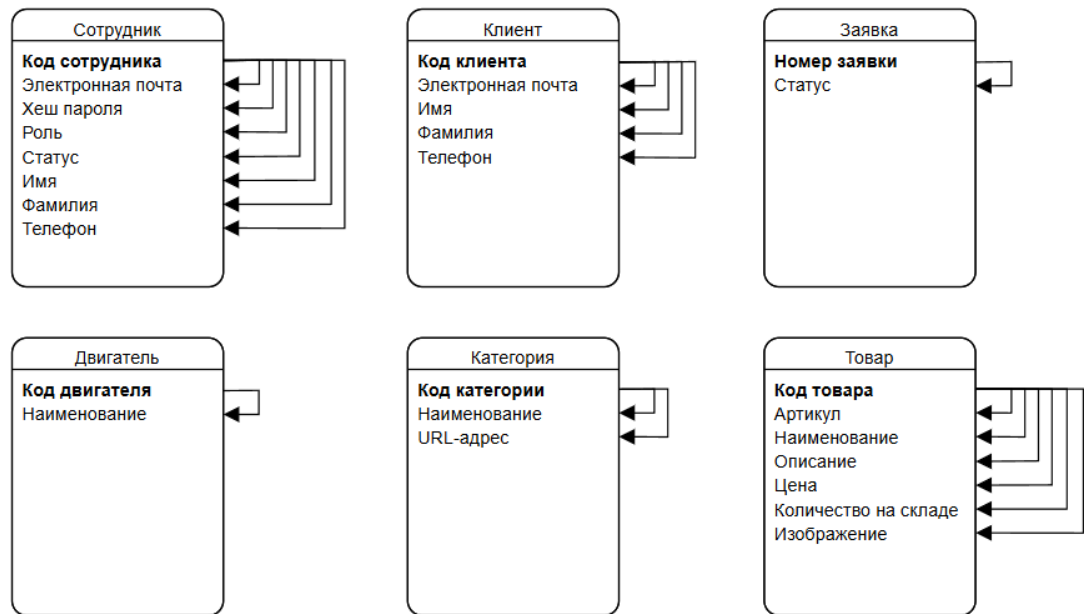


Рисунок 15 – Итоговые сущности в 2НФ

Для атрибутов «Статус» и «Роль» не планируется создавать отдельные таблицы, так как их расширение не планируется, поэтому они будут реализованы в виде строк, которые будут задаваться через ENUM структуры на уровне запросов к БД.

Третья нормальная форма (3НФ) – это уровень нормализации, который требует, чтобы все атрибуты в отношении зависели только от ключа и не зависели от других атрибутов.

3НФ помогает избежать избыточности данных и устраняет аномалии вставки, обновления и удаления данных. Он способствует более эффективному хранению данных и обработке запросов, а также обеспечивает более гибкую структуру для модификации базы данных.

Для достижения 3НФ необходимо выполнить следующие условия:

- отношение должно быть уже в 2НФ;
- не должно быть транзитивных функциональных зависимостей. Транзитивная функциональная зависимость возникает, когда один атрибут зависит

от другого атрибута, который, в свою очередь, зависит от ключа. Если такая зависимость существует, то атрибуты, которые являются промежуточными в этой зависимости, должны быть выделены в отдельное отношение.

Все сущности приведены к 3НФ.

#### 2.3.4 Реляционная модель данных

Отображение полученной концептуально-информационной модели на реляционную модель включает представление ключевых элементов взаимосвязанных записей в отношениях. Отношения в реляционной модели представляются в виде таблиц, где каждая запись представляет собой строку, а каждый атрибут – столбец.

Одной из важнейших аспектов реляционной модели данных является возможность установления связей между различными сущностями базы данных. Родительские и дочерние сущности, или таблицы, могут быть связаны между собой через ключи, которые задаются определенными атрибутами этих таблиц.

Сущность, являющаяся родительской, имеет определенный набор атрибутов, включающий в себя ключевой атрибут. Дочерняя сущность, в свою очередь, содержит атрибут, который является ключом и одновременно ссылается на родительскую таблицу. Таким образом, между родительскими и дочерними сущностями устанавливается связь, позволяющая обеспечить целостность и согласованность данных в базе данных.

Связь между родительским и дочерним отношениями может быть один к одному, один ко многим или многие ко многим. – один к одному. Это означает, что одной записи из первой таблицы соответствует только одна запись из второй таблицы (между первичными ключами);

– один ко многим. Это означает, что одной записи из первой таблицы соответствует одна или несколько записей из другой таблицы (между первичным и внешним ключом, между первичным и первичным составным);

– многие ко многим: возможно в случаях, когда обязательно выполняются следующие два условия:

- одной записи из первой таблице соответствует одна или несколько записей из другой таблицы;
- одной записи второй таблицы соответствует одна или несколько записей из первой таблицы.

Сущность «Клиент» является исходной (родительской), а сущность «Заявка» является порождённой (дочерней). Связь «Один-Ко-Многим», следовательно, ключ исходной (родительской) сущности добавляем в порождённую (дочернюю), что показано на рисунке 16.



Рисунок 16 –Связь «Клиент – Заявка»

Результат анализа связи показан на рисунке 17.

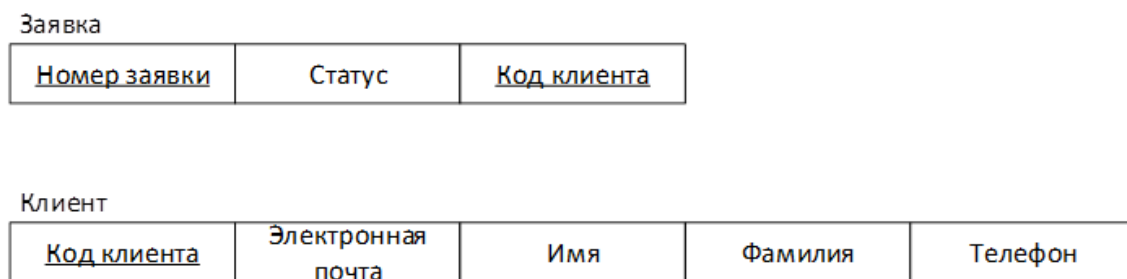


Рисунок 17 – Результат анализа связи «Клиент – Заявка»

Сущность «Сотрудник» является исходной (родительской), а сущность «Заявка» является порождённой (дочерней). Связь «Один-Ко-Многим», следовательно, ключ исходной (родительской) сущности добавляем в порождённую (дочернюю), что показано на рисунке 18.



Рисунок 18 –Связь «Сотрудник – Заявка»

Результат анализа связи показан на рисунке 19.

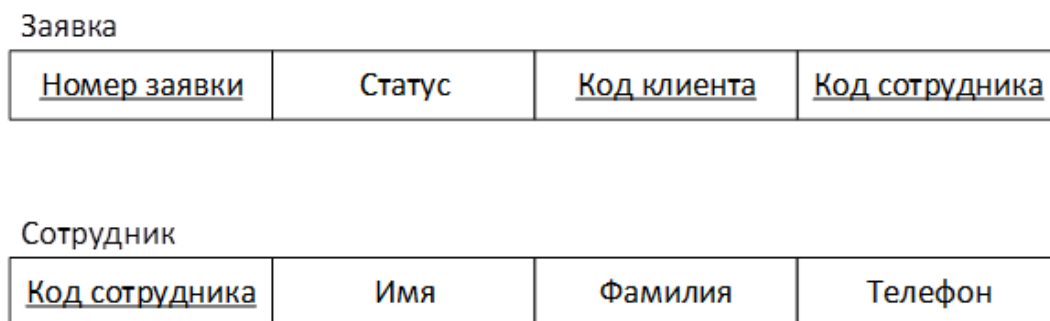


Рисунок 19 – Результат анализа связи «Клиент – Заявка»

Сущность «Двигатель» является исходной (родительской), а сущность «Товар» является порождённой (дочерней). Связь «Один-Ко-Многим» (с опциональностью), следовательно, ключ исходной сущности добавляем в порождённую, что показано на рисунке 20.



Рисунок 20 – Связь «Двигатель – Товар»

Результат анализа связи показан на рисунке 21.

|                   |                      |                      |          |      |
|-------------------|----------------------|----------------------|----------|------|
| Товар             |                      |                      |          |      |
| <u>Код товара</u> | Артикул              | Наименование         | Описание | Цена |
| Изображение       | Количество на складе | <u>Код двигателя</u> |          |      |

|                      |              |
|----------------------|--------------|
| Двигатель            |              |
| <u>Код двигателя</u> | Наименование |

Рисунок 21 – Результат анализа связи «Двигатель – Товар»

В отношении между сущностями «Товар» и «Категория» (через ассоциативную сущность «Товары\_Категории») установлена связь «Многие-Ко-Многим». Ключи обеих сущностей добавляются в порождённую (дочернюю) сущность «Товар\_Категория», что показано на рисунке 22.

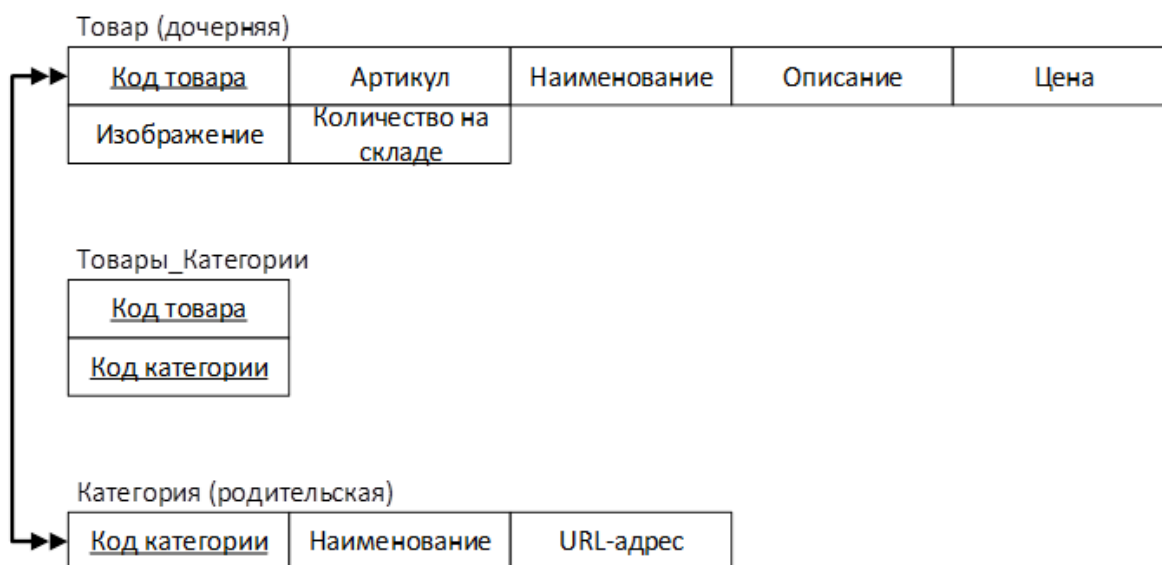


Рисунок 22 – Связь «Товар – Категория»

Результат анализа связи показан на рисунке 23.

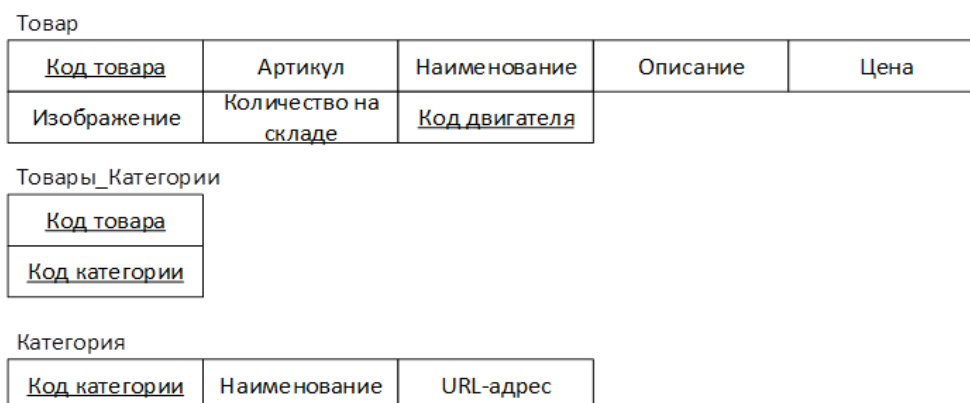


Рисунок 23 – Результат анализа связи «Товар – Категория»

Для отношения «Товар – Заявка» установлена связь «Многие-Ко-Многим», поэтому необходимо создать ассоциативную сущность «Товары\_в\_заявке», но в неё также стоит включить атрибут «Количество», чтобы сократить количество одинаковых записей в будущей таблице.

Сущность «Заявка» является исходной (родительской), а сущность «Товары\_в\_заявке» – порождённой (дочерней). Связь «Один-Ко-Многим» (идентифицирующая), где ключ родительской сущности входит в составной первичный ключ дочерней, что показано на рисунке 24.

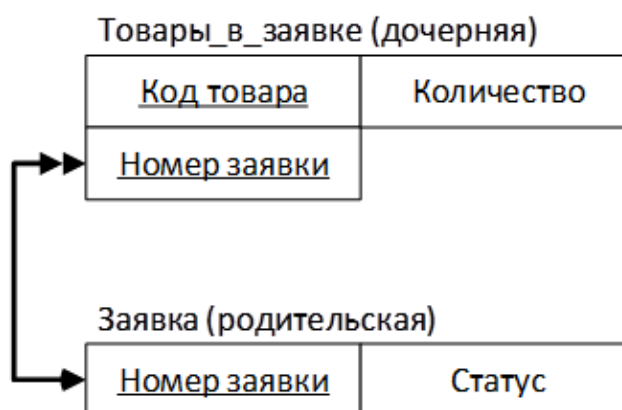


Рисунок 24 – Связь «Заявка – Товары\_в\_заявке»

Сущность «Товар» является исходной (родительской), а сущность «Товары\_в\_заявке» – порождённой (дочерней). Связь «Один-Ко-Многим» (идентифицирующая), где ключ родительской сущности входит в составной первичный ключ дочерней, что показано на рисунке 25.



Рисунок 25 – Связь «Товар – Товары\_в\_заявке»

Результат анализа связи «Товар – Заявка» показан на рисунке 26.



Рисунок 26 – Результат анализа связи «Товар – Заявка»

В результате отображения концептуально-инфологической модели на реляционную получается совокупность отношений реляционной модели. Конечные отношения представлены на рисунке 27.

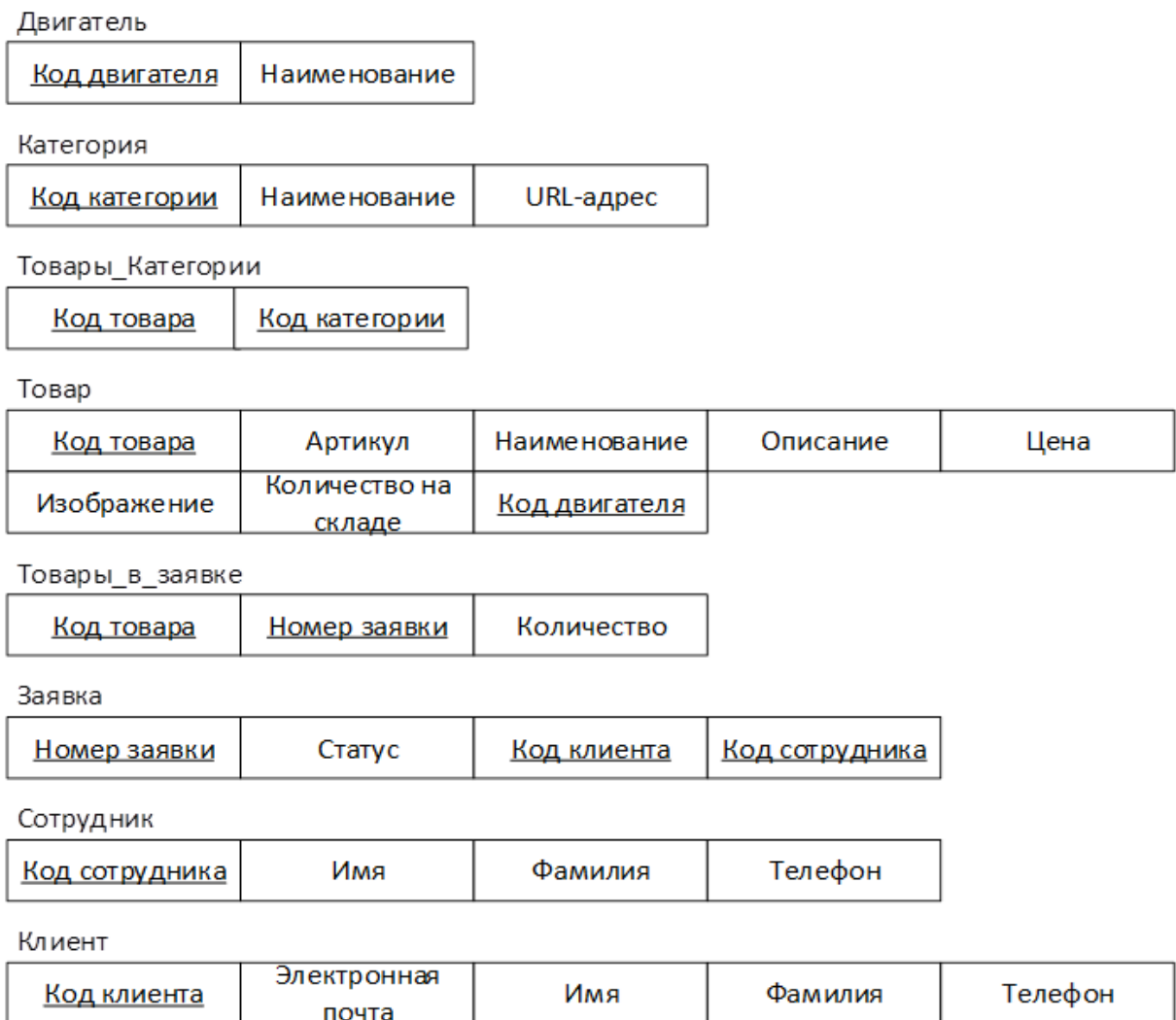


Рисунок 27 – Реляционная модель базы данных

### 2.3.5 Логическая модель данных

Логическая модель данных даёт представление структуры и отношений между данными в БД на уровне концептуального проектирования. ЛМД описывает сущности (таблицы), их атрибуты (столбцы) и связи между сущностями. Также помогает понять, каким образом данные будут храниться и взаимодействовать друг с другом.

При проектировании логической модели данных учитываются требования к базе данных, в том числе ее функциональные и нефункциональные характеристики. От правильного выбора сущностей и атрибутов зависит эффективность и надежность работы базы данных.

Логическая модель данных позволяет определить структуру и связи между данными без привязки к конкретному физическому хранилищу или программному обеспечению. Это обеспечивает гибкость и удобство внесения изменений в базу данных в дальнейшем.

Диаграмма IDEF1X (Integrated Definition for Information Modeling) – это графическое представление логической модели данных, которое используется для описания структуры данных, их связей и ограничений. Диаграмма IDEF1X включает в себя сущности (таблицы), их атрибуты (поля) и связи между сущностями. На диаграмме IDEF1X каждая сущность представлена прямоугольником, внутри которого указывается название сущности, и её атрибуты. Связи между сущностями обозначаются линиями, сопровождающимися указанием типа связи (один-к-одному, один-ко-многим и т. д.) и ограничениями на связи.

Диаграмма IDEF1X представлена на рисунке 28.

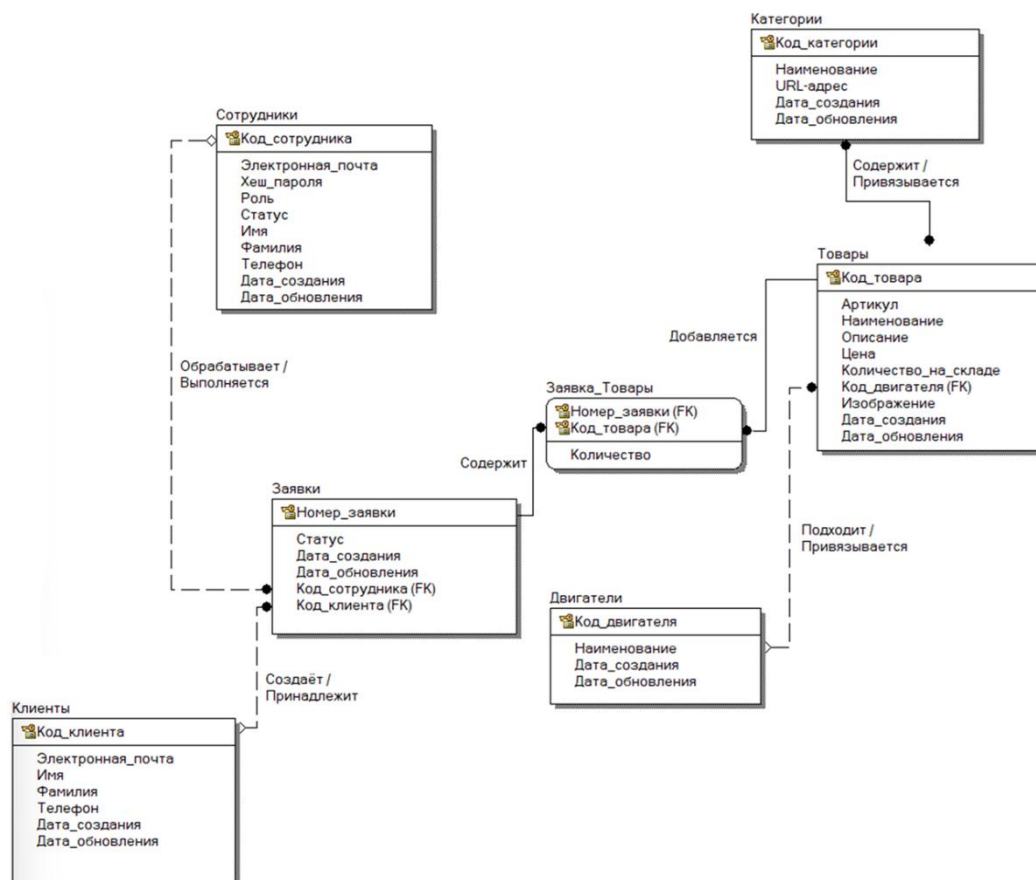


Рисунок 28 – Логическая модель базы данных в нотации IDEF1X

Стоит заметить, что связь «Товары – Заявки» имеет характер отношения «многие ко многим». В связи с этим возникает необходимость создания новой таблицы, в которой будут зафиксированы ключевые атрибуты исходных таблиц. Полученная таблица для этого отношения представлена в таблице 7.

Таблица 7 – Описание таблицы связи заявок с товарами

| Название атрибута | Описание атрибута | Тип данных | Диапазон значений | Пример атрибута |
|-------------------|-------------------|------------|-------------------|-----------------|
| Номер заявки (FK) |                   | Числовой   | >0                | 1               |
| Код товара (FK)   |                   | Числовой   | >0                | 1               |
| Количество        |                   | Числовой   | >0                | 10              |

Таким образом, логическая модель базы данных полностью описывает логические связи и отношения между сущностями и отображает то, что все сущности приведены к третьей нормальной форме.

#### 2.3.6 Физическая модель данных

Физическое проектирование базы данных представляет собой процесс, в ходе которого устанавливаются структура и организация данных с учетом физических характеристик их хранения. На данном этапе осуществляется выбор типов данных, которые будут применяться для каждого атрибута. Физическое определение атрибутов приведено в таблицах 8-14.

Таблица 8 – Описание таблицы сотрудников

| Название атрибута | Тип данных   | Условия    | Формат данных | Индексация  |
|-------------------|--------------|------------|---------------|-------------|
| Код сотрудника    | Целое число  | >0         | Int           | Primary Key |
| Электронная почта | Строка (255) | –          | Text(255)     | –           |
| Хеш пароля        | Строка (255) | –          | Text(255)     | –           |
| Роль              | Строка (20)  | –          | Text(20)      | –           |
| Статус            | Логический   | false/true | Boolean       | –           |
| Имя               | Строка (50)  | –          | Text(50)      | –           |
| Фамилия           | Строка (50)  | –          | Text(50)      | –           |
| Телефон           | Строка (20)  | –          | Text(50)      | –           |
| Дата создания     | Дата/Время   | –          | DateTime      | –           |
| Дата обновления   | Дата/Время   | –          | DateTime      | –           |

Таблица 9 – Описание таблицы клиентов

| Название атрибута | Тип данных   | Условия | Формат данных | Индексация  |
|-------------------|--------------|---------|---------------|-------------|
| Код клиента       | Целое число  | >0      | Int           | Primary Key |
| Электронная почта | Строка (255) | –       | Text(255)     | –           |
| Имя               | Строка (50)  | –       | Text(50)      | –           |
| Фамилия           | Строка (50)  | –       | Text(50)      | –           |
| Телефон           | Строка (20)  | –       | Text(20)      | –           |
| Дата создания     | Дата/Время   | –       | DateTime      | –           |
| Дата обновления   | Дата/Время   | –       | DateTime      | –           |

Таблица 10 – Описание таблицы заявок

| Название атрибута | Тип данных  | Условия | Формат данных | Индексация  |
|-------------------|-------------|---------|---------------|-------------|
| Номер заявки      | Целое число | >0      | Int           | Primary Key |
| Код клиента       | Целое число | >0      | Int           | Foreign Key |
| Код сотрудника    | Целое число | >0      | Int           | Foreign Key |
| Статус            | Строка (20) | –       | Text(20)      | –           |
| Дата создания     | Дата/Время  | –       | DateTime      | –           |
| Дата обновления   | Дата/Время  | –       | DateTime      | –           |

Таблица 11 – Описание таблицы двигателей

| Название атрибута | Тип данных   | Условия | Формат данных | Индексация  |
|-------------------|--------------|---------|---------------|-------------|
| Код двигателя     | Целое число  | >0      | Int           | Primary Key |
| Наименование      | Строка (100) | –       | Text(100)     | –           |
| Дата создания     | Дата/Время   | –       | DateTime      | –           |
| Дата обновления   | Дата/Время   | –       | DateTime      | –           |

Таблица 12 – Описание таблицы категорий

| Название атрибута | Тип данных   | Условия | Формат данных | Индексация  |
|-------------------|--------------|---------|---------------|-------------|
| Код категории     | Целое число  | >0      | Int           | Primary Key |
| Наименование      | Строка (100) | –       | Text(100)     | –           |
| URL-адрес         | Строка (100) | –       | Text(100)     | –           |
| Дата создания     | Дата/Время   | –       | DateTime      | –           |
| Дата обновления   | Дата/Время   | –       | DateTime      | –           |

Таблица 13 – Описание таблицы товаров

| Название атрибута    | Тип данных     | Условия  | Формат данных | Индексация  |
|----------------------|----------------|----------|---------------|-------------|
| Код товара           | Целое число    | $>0$     | Int           | Primary Key |
| Артикул              | Строка (50)    | –        | Text(50)      | –           |
| Наименование         | Строка (255)   | –        | Text(255)     | –           |
| Описание             | Текст          | –        | Text          | –           |
| Цена                 | Число с пл. т. | $\geq 0$ | Float         | –           |
| Количество на складе | Целое число    | $\geq 0$ | LongInt       | –           |
| Код двигателя        | Целое число    | $>0$     | Int           | Foreign Key |
| Изображение          | Строка (255)   | –        | Text(255)     | –           |
| Дата создания        | Дата/Время     | –        | DateTime      | –           |
| Дата обновления      | Дата/Время     | –        | DateTime      | –           |

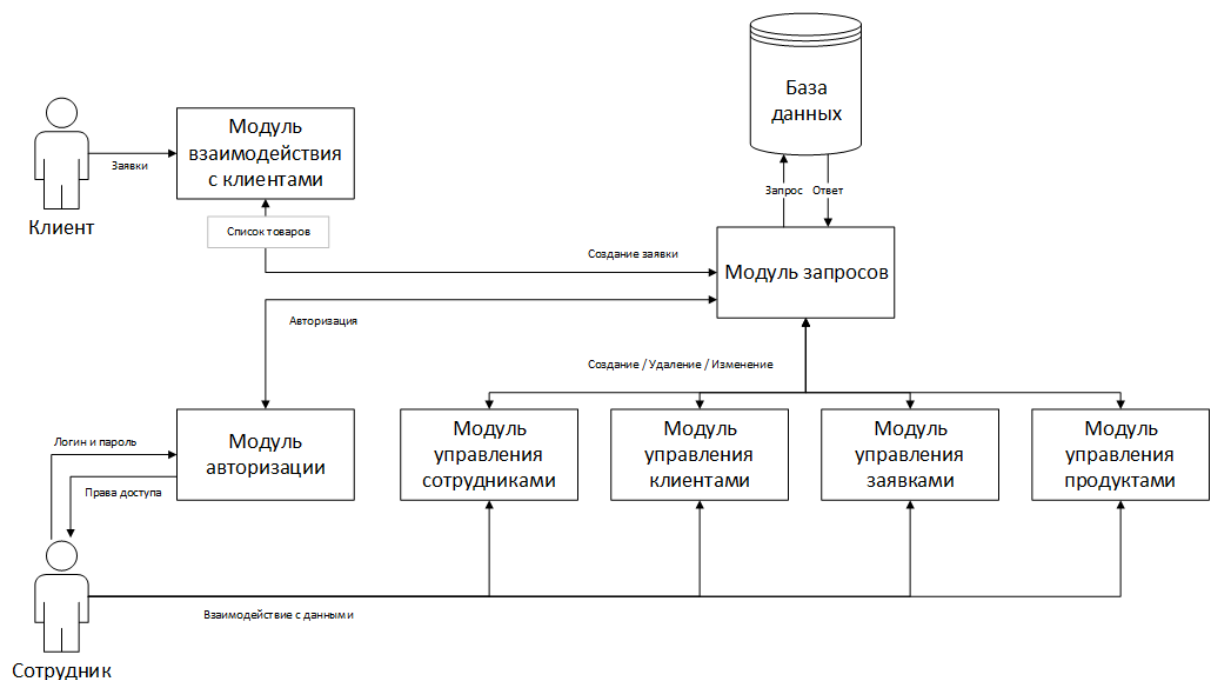
Таблица 14 – Описание вспомогательной таблицы связи заявок и товаров

| Название атрибута | Тип данных | Условия | Формат данных | Индексация  |
|-------------------|------------|---------|---------------|-------------|
| Номер заявки      | Числовой   | $>0$    | Int           | Foreign Key |
| Код товара        | Числовой   | $>0$    | Int           | Foreign Key |
| Количество        | Числовой   | $>0$    | Int           |             |

Физическая модель базы данных в нотации IDEF1X представлена на рисунке 29.



представляет собой схему, показывающую взаимосвязи между различными компонентами приложения и их роль в общем процессе обработки данных. Такая структура позволяет четко выделить функциональные блоки, такие как каталог продуктов, корзина для покупателя и административная панель для сотрудников, а также продемонстрировать их взаимодействие. Это облегчает понимание архитектуры системы и ее функциональных возможностей, а также служит основой для дальнейшей разработки и тестирования.



Анализ функциональных требований информационной системы показывает, что система будет состоять из нескольких ключевых модулей. Основные модули включают в себя модуль авторизации для сотрудников, модуль управления сотрудниками, модуль управления продуктами и модуль управления заявками. Для обеспечения полной функциональности и удобства взаимодействия пользователей с системой, каждый интерфейс будет предназначен для отображения и редактирования информации, поэтому число интерфейсов,

предоставляющих пользователям доступ к функционалу, будет больше числа модулей.

#### 2.4.2 Архитектура информационной системы

Финальным этапом проектирования информационной системы становится разработка архитектуры, которая представляет собой структурное описание системы, определяющее ее компоненты, взаимосвязи между ними и принципы взаимодействия. Архитектура информационной системы служит основой для ее реализации и функционирования, обеспечивая ясное понимание структуры и поведения системы.

В разработанной информационной системе реализована многоуровневая архитектура, представленная на рисунке 31 предполагающая разделение системы на несколько уровней, каждый из которых выполняет свои специфические функции и задачи. Архитектура включает три уровня:

- Уровень представления: отвечает за взаимодействие пользователей с системой, включая графические элементы, кнопки, формы и элементы навигации, доступные как для клиентов, так и для сотрудников.
- Уровень бизнес-логики: представляет собой ядро системы, где осуществляется обработка информации, включая авторизацию сотрудников, управление заявками и продуктами. На этом уровне реализуются основные сценарии работы, такие как добавление товаров в корзину и оформление заявок.
- Уровень хранения данных: отвечает за сохранение информации в базе данных, включая данные сотрудников, клиентов, заявки и продукты вместе с их взаимосвязями. Этот уровень обеспечивает целостность и безопасность хранимых данных.

Такая архитектура позволяет эффективно управлять взаимодействиями между пользователями и системой, обеспечивая высокий уровень производительности и удобства для конечных пользователей.

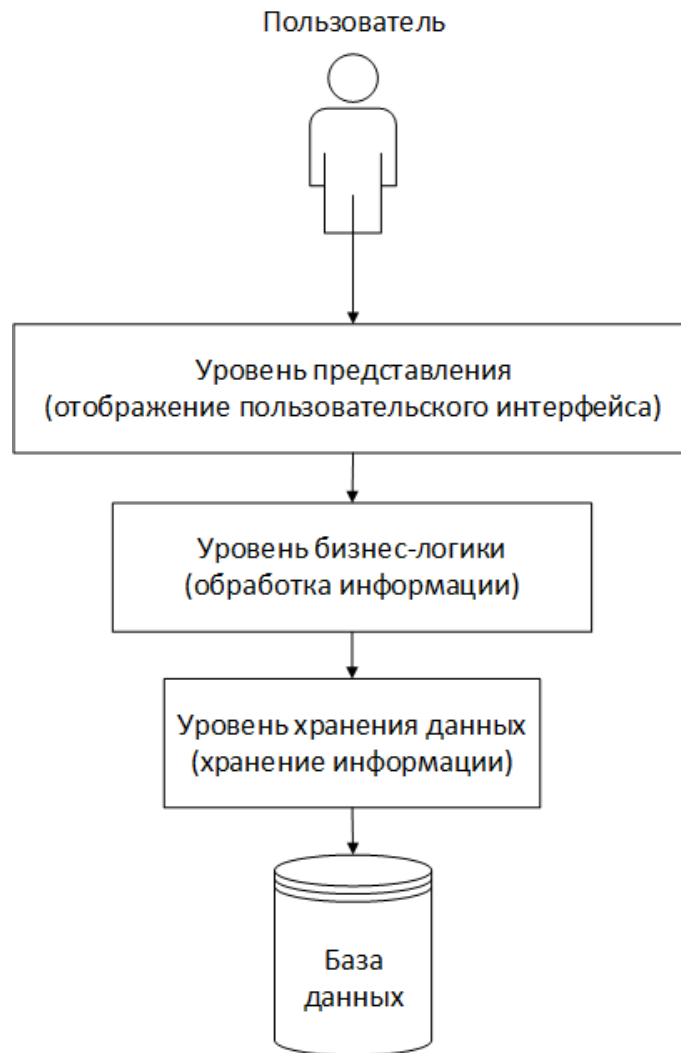


Рисунок 31 – Архитектура информационной системы

Архитектура системы предусматривает два уровня доступа: администраторы (ADMIN) и рядовые сотрудники (USER). Ролевая модель реализована через JWT-токены, содержащие информацию о правах пользователя в зашифрованном виде. Middleware-слой анализирует принадлежность маршрута к защищенным зонам и проверяет наличие требуемой роли через декодирование токена. Для администраторов доступны операции CRUD для всех сущностей системы, включая управление правами других пользователей. Обычные сотрудники ограничены в возможностях модификации системных настроек и просмотра конфиденциальных данных. Механизм блокировки учетных записей (поле *enabled*) позволяет временно деактивировать доступ без удаления записей из базы.

Ядро модуля реализовано через транзакционные операции с гарантией целостности данных. При создании заказа выполняется атомарная проверка остатков товаров, резервирование позиций и генерация связанных сущностей (покупатель, товарные позиции). Статусная модель жизненного цикла заказа включает этапы «Новый», «В обработке», «Отправлен», «Завершен» и «Отменен», с автоматическим пересчетом складских остатков при изменении состояния. Для сложных аналитических запросов (расчет выручки, динамика продаж) используются оптимизированные SQL-запросы с оконными функциями. Интеграция с модулем покупателей реализована через систему автодополнения email при создании заявок.

#### 2.4.3 Проектирование пользовательского интерфейса модулей

Пользовательский интерфейс будет построен на компонентах Material UI с кастомизированной темой, обеспечивающей единообразие визуального оформления. Маршрутизация будет организована по принципу RESTful с разделением на административные и публичные зоны. Для табличных данных будет реализована виртуализация строк и пагинация с серверной обработкой фильтров. Формы ввода смогут использовать многоуровневую валидацию, например, клиентская проверка форматов (email, телефон) через регулярные выражения, серверная валидация уникальности артикулов и URL-идентификаторов, бизнес-проверки остатков товаров при оформлении заказов.

Система уведомлений будет реализована через Toast-сообщения с цветовой индикацией статусов операций. Для критически важных действий (удаление записей) предусмотрена двухэтапная проверка через модальные диалоги подтверждения. В аналитический раздел будет интегрирована библиотека Recharts для визуализации данных через интерактивные графики и диаграммы, данные для которых формируются агрегационными запросами к OLAP-структурам.

Пользовательский интерфейс (UI) CRM-системы спроектирован с учетом удобства использования и интуитивной навигации. Он реализован с

использованием React-компонентов и стилей CSS, что обеспечивает адаптивность и визуальную привлекательность.

Техническая реализация разделена на слои: компоненты интерфейса размещены в отдельных специальных директориях, бизнес-логика вынесена в кастомные хуки, а взаимодействие с API организовано через службы-сервисы. Для обработки асинхронных запросов используются индикаторы загрузки, а ввод специализированных данных (например, телефонных номеров) осуществляется через компоненты с масками.

## **2.5 Проектирование системы безопасности**

### **2.5.1 Аутентификация пользователей**

Механизм аутентификации реализован на базе фреймворка NextAuth с использованием стратегии JSON Web Tokens (JWT). Система предусматривает двухфакторную верификацию: на клиентском уровне выполняется проверка формата учетных данных, на серверном – поиск сотрудника в базе данных через Prisma ORM с последующим сравнением хешей паролей при помощи алгоритма bcrypt. Успешная авторизация инициирует формирование JWT-токена с 30-дневным сроком действия, содержащего минимальный набор полей (идентификатор сотрудника, роль, email) для снижения рисков утечки данных.

Сессионные куки защищены флагами *httpOnly* и *SameSite=lax*, исключающими возможность перехвата через XSS-атаки. В production-окружении активируется режим *Secure* с обязательным использованием HTTPS-соединения. Для предотвращения brute-force атак реализован механизм прогрессивной задержки – при пяти последовательных неудачных попытках входа система увеличивает время ответа экспоненциально от 1 до 30 секунд.

### **2.5.2 Управление правами доступа**

Ролевая модель доступа построена на принципе минимальных привилегий (POLP) с двумя базовыми ролями: администратор (ADMIN) и пользователь (USER). Маршрутизация защищена middleware-слоем, который анализирует JWT-токен и сверяет список разрешенных ролей для каждого API-эндпоинта.

Для динамических маршрутов реализована система ACL (Access Control List), сопоставляющая URL-паттерны с матрицей прав доступа.

Критически важные операции (удаление записей, изменение ролей) требуют дополнительной проверки через модальные диалоги подтверждения на клиентской стороне. Серверная валидация прав реализована через декораторы запросов в Prisma, гарантирующие отсутствие прямого доступа к raw-запросам. Для защиты от вертикальной эскалации привилегий все мутационные операции содержат дополнительную проверку `session.user.role = «ADMIN»` в resolver-функциях.

Интеграция с фронтендом осуществляется через HOC (Higher-Order Component), осуществляющий перенаправление неавторизованных пользователей на страницу входа с сохранением исходного URL.

## **2.6 Проектирование модуля взаимодействия с клиентами**

Страницы каталога реализуют двухуровневую систему навигации: категории и подкатегории товаров. Для обеспечения производительности данные будут загружаться асинхронно с использованием стратегии ISR (Incremental Static Regeneration), что позволяет актуализировать контент без полного перестроения страницы. Карточки товаров проектируются с адаптивным дизайном, обеспечивающим корректное отображение на различных устройствах. Система предусматривает интеллектуальную подгрузку изображений с отложенной загрузкой (lazy loading) и скелетонами для улучшения пользовательского опыта при медленном соединении. Фильтрация реализуется через параметры URL, что сохраняет возможность глубоких ссылок и истории просмотров.

Корзина проектируется как клиент-серверное решение с синхронизацией состояния через сессионные cookie, защищённые криптографическим sealing механизмом. Для минимизации нагрузки на БД используется оптимистичное обновление интерфейса: изменения количества товаров сразу отражаются в UI с последующей асинхронной фиксацией в хранилище. Валидация остатков товаров будет реализована через проверку доступного количества на сервере перед каждым изменением. Система предусматривает трёхэтапную

верификацию данных пользователя: синтаксический анализ полей, проверку форматов и бизнес-логику доступности товаров. Для предотвращения потери данных используем стратегию локального сохранения состояния при возникновении ошибок соединения.

Оформление заказа проектируется как атомарная транзакция, объединяющая обновление остатков товаров, создание записи заказа и привязку клиента. Для обеспечения консистентности данных применяется паттерн Unit of Work с изоляцией операций на уровне базы данных. Интеграция с email-сервисом реализуется через асинхронную очередь задач с повторными попытками отправки. Система генерирует уникальные веб-ссылки для отслеживания статуса заказа, защищённые комбинацией идентификатора заказа и email клиента. Архитектура модуля разделяет ответственности: валидация входных данных, бизнес-логика обработки заказа и формирование выходных представлений будут реализованы в отдельных слоях для обеспечения тестируемости и масштабируемости.

## 3 РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ

### 3.1 Реализация клиентской части

Клиентская часть веб-приложения реализована с использованием фреймворка Next.js 15.3, обеспечивающего гибридный рендеринг и совместимость с экосистемой React. Архитектура приложения построена на принципах модульности и компонентного подхода. Базовый каркас приложения инициализируется в файле, который является точкой входа в приложение, где осуществляется интеграция критически важных модулей: система аутентификации NextAuth, библиотека Material UI и механизм глобального состояния Toast-уведомлений [2].

Типографическая система приложения настроена с использованием шрифта Roboto через механизм оптимизированной подгрузки Google Fonts. Кастомизация визуального оформления выполнена посредством создания темы с помощью Material UI, где заданы параметры цветовой схемы, адаптивные точки и глобальные переопределения компонентов.

Система переменных окружения организована через env-файлы с разделением секретных ключей и технических параметров. Интеграция Prisma ORM обеспечена через скрипты миграции и сидинга данных, что позволяет воспроизводить тестовые сценарии с заранее заполненными учётными записями администратора.

Оптимизация производительности достигнута за счёт:

- серверного рендеринга критически важных страниц (каталог, CRM-панель);
- динамического импорта тяжеловесных компонентов (графики, редакторы);
- кэширования запросов через библиотеку SWR;
- настройки tree-shaking для Material UI компонентов.

Архитектура клиентской части предусматривает расширяемость – система абсолютных импортов и модульная структура папок позволяют

масштабировать функционал без нарушения существующей логики. Интеграция TypeScript-типизации для всех внешних зависимостей обеспечивает строгую типизацию и предотвращает ошибки во время работы программы.

### **3.2 Разработка серверной логики**

Архитектура серверной части реализована по принципам RESTful API с использованием Next.js API Routes, обеспечивающих строгое разделение клиентских и серверных процессов. Ядром системы выступает многоуровневая модель обработки запросов, включающая промежуточное ПО для контроля доступа, валидации данных и обработки исключений.

Система аутентификации построена на основе NextAuth со специально настроенным провайдером учетных данных, интегрированным с механизмом сессий на базе JWT-токенов. Авторизация реализована через ролевую модель, где каждый API-эндпоинт ассоциирован с разрешенными ролями (ADMIN, USER). Middleware-слой анализирует заголовки запросов, проверяет валидность токенов и перенаправляет неавторизованные обращения на страницу входа, сохраняя параметры ошибок для последующего отображения.

Обработка бизнес-логики организована через транзакционные операции Prisma ORM, гарантирующие атомарность критически важных процессов. При оформлении заказов реализован двухэтапный механизм: предварительная проверка остатков товаров с последующим атомарным обновлением данных в рамках ACID-транзакции [3]. Для предотвращения гонки запросов при параллельных запросах используется оптимистичная блокировка на уровне приложения.

Система управления данными включает специализированные API-эндпоинты, охватывающие полный цикл CRUD-операций для сущностей предметной области. Реализована расширенная валидация входных параметров с использованием специальных валидаторов, проверяющих целостность ссылок между таблицами и соответствие бизнес-правилам. Например, при изменении категории товара осуществляется проверка существования связей перед выполнением операций обновления.

Интеграция с базой данных реализована через абстракцию Prisma Client, обеспечивающую безопасность типов в запросах и автоматическую генерацию DTO-объектов. Для сложных аналитических запросов (расчет выручки, статистика по статусам заказов) задействованы RAW-запросы SQL, оптимизированные под специфику SQLite. Репликация часто используемых данных (категории, двигатели) осуществляется через механизм кэширования в памяти на уровне API.

Безопасность данных обеспечена многоуровневой системой защиты, включающей в себя хеширование паролей по алгоритму bcrypt, санитизацию SQL-запросов через параметризованные выражения, валидацию входных параметров с помощью регулярных выражений, обработку ошибок через глобальный перехватчик с маскировкой чувствительной информации и ролевую модель доступа с разделением административных и пользовательских операций

Для пагинации результатов реализован универсальный механизм, принимающий параметры *page* и *limit*, с автоматическим расчетом метаданных (общее количество записей, число страниц). Фильтрация данных поддерживает комбинированные условия по множеству полей, включая диапазоны цен, текстовый поиск и фильтры по связанным сущностям.

Статистический модуль генерирует аналитические отчеты в реальном времени, используя агрегационные запросы к базе данных. Расчет ключевых показателей (общая выручка, динамика продаж за период) оптимизирован через предварительную материализацию часто запрашиваемых данных. Визуализация статистики реализована через интеграцию с React библиотекой Recharts [4], где серверная часть предоставляет структурированные данные в формате, оптимальном для построения графиков.

Обработка ошибок стандартизирована через единый формат ответов с кодами состояния HTTP и детализированными сообщениями. Для критических сценариев (недостаток товарных остатков, конфликты уникальности) предусмотрены специализированные коды ошибок с автоматическим

уведомлением администратора системы. Логирование операций осуществляется как на уровне приложения, так и через встроенные механизмы мониторинга базы данных.

### **3.3 Реализация базы данных**

Архитектура базы данных реализована на основе реляционной модели с использованием SQLite в качестве основной СУБД для среды разработки. Выбор SQLite обусловлен требованиями к простоте развертывания и минимальной конфигурации – хранение данных в едином файле позволяет отказаться от выделенного сервера баз данных, сохраняя при этом полноценную поддержку ACID-транзакций. Конфигурация подключения задана через переменную окружения, что обеспечивает гибкость при переходе на промышленные СУБД (PostgreSQL/MySQL) в production-окружении.

Структура данных определена через декларативную схему Prisma, описывающую взаимосвязанные сущности. Нормализация до третьей нормальной формы достигнута за счет декомпозиции предметной области на изолированные таблицы: «Сотрудники», «Клиенты», «Заявки», «Товары», «Двигатели», «Категории» и связующую таблицу «Заявки\_Товары». Отношения «один-ко-многим» реализованы через внешние ключи (например, *customerId* в таблице «Заявки»), а для связи «многие-ко-многим» между заказами и товарами введена промежуточная таблица с составным первичным ключом.

Миграция базы данных осуществляется через систему версионирования Prisma, генерирующую SQL-скрипты на основе изменений в схеме. Инициализация тестовых данных выполняется скриптом сидинга, который создает демонстрационных тестовых сотрудников, товары с привязкой к случайным двигателям, клиентов и заявки с реалистичными связями. Алгоритм сидинга включает в себя хеширование паролей администраторов по алгоритму bcrypt, генерацию псевдослучайных артикулов товаров по маске, распределение товаров между заказами с учетом ограничений остатков, автоматическую привязку категорий к товарам через механизм транзакций.

Интеграция с приложением реализована через Prisma Client – это ORM с безопасностью типов данных, генерируемый на основе схемы. Экземпляр клиента инициализируется в отдельном модуле, обеспечивающем единую точку входа для всех операций с данными. Оптимизация производительности достигнута за счет пакетной обработки запросов, использования индексов для часто фильтруемых полей, кэширования подключений к базе данных, оптимистичных блокировок при конкурентных обновлениях.

Система безопасности включает в себя валидацию форматов данных на уровне схемы (уникальность *email*, ограничения длины строк), автоматическое обновление временных меток (*createdAt*, *updatedAt*), каскадное удаление связанных записей через механизм каскадного удаления, шифрование чувствительных данных (например, пароли) перед сохранением.

Для отладки и анализа запросов активирован режим логирования SQL-операций в development-окружении [5]. Репликация данных между средами выполняется через механизм дампов, гарантирующий консистентность структуры на всех этапах разработки.

### **3.4 Разработка CRM-панели**

CRM-панель представляет собой многоуровневую систему управления бизнес-процессами, реализованную на базе фреймворка Next.js. Архитектура решения построена по модульному принципу, где каждый функциональный блок обеспечивает выполнение специфических задач, интегрируясь в единый пользовательский интерфейс через систему ролевого доступа и стандартизированные API-эндпоинты [6].

Модуль авторизации и аутентификации реализован с использованием NextAuth, обеспечивающего интеграцию JWT-стратегии с 30-дневным сроком действия токенов. Механизм аутентификации основан на Credentials-провайдере, выполняющем поиск пользовательских записей в базе данных через Prisma ORM с последующей верификацией хешей паролей средствами библиотеки bcrypt. Успешная авторизация инициирует формирование кастомного JWT-токена, содержащего идентификатор сотрудника, роль и метаданные

сессии. Безопасность передачи данных обеспечена настройками куки-файлов: флаги *httpOnly* и *SameSite=lax* исключают возможность XSS-атак, а в production-окружении активируется режим HTTPS-only. Конфигурация системы централизована через переменные окружения, что упрощает управление секретами в разных средах развёртывания.

Доступ к CRM-модулю обеспечивается через систему аутентификации, показанный на рисунке 32, с использованием компонент-обёртки, который контролирует переходы между страницами и перенаправляет незарегистрированных пользователей на форму входа.

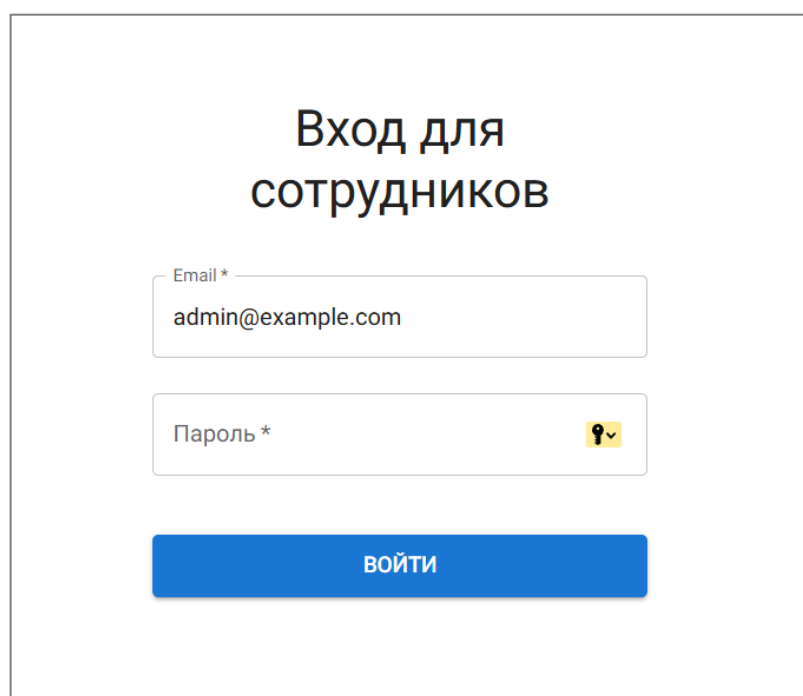


Рисунок 32 – Страница авторизации в админ панели

Модуль управления заявками предоставляет инструменты для обработки заказов через интерфейс, включая фильтрацию по статусам, детализацию состава товаров и изменение жизненного цикла заявок. Бизнес-логика реализована через транзакционные операции Prisma, гарантирующие атомарность обновления связанных сущностей. Для аналитических отчётов задействованы RAW-SQL-запросы с оконными функциями, позволяющие рассчитывать ключевые метрики (выручка, динамика продаж) в реальном времени.

Модуль управления категориями обеспечивает создание иерархических структур товарных групп с валидацией уникальности URL-идентификаторов на уровне API. Реализован механизм каскадной проверки связей перед удалением категорий, исключающий потерю данных. Для массовых операций импорта/экспорта используется потоковая обработка CSV-файлов с ограничением нагрузки на базу данных через пакетные транзакции.

Модуль управления товарами интегрирует CRUD-функционал для работы с продуктами, включая многоуровневую фильтрацию по двигателям, категориям и ценовым диапазонам. Валидация полей товаров осуществляется гарантирует соответствие внутренним стандартам предприятия. Автоматический пересчёт складских остатков реализован через триггеры на уровне приложения, синхронизирующие данные между заказами и товарными позициями.

Модуль управления сотрудниками предоставляет администраторам инструменты для настройки ролевой модели (ADMIN, USER) с гранулярным разграничением прав. Реализована двухэтапная валидация вводимых данных: проверка формата электронной почты через регулярные выражения и верификация уникальности записей средствами Prisma ORM. В будущем можно реализовать механизм сброса паролей, который включает генерацию временных токенов с отправкой на электронную почту и последующим хешированием новых учётных данных через bcrypt.

Модуль управления покупателями обеспечивает ведение клиентской базы с полнотекстовым поиском по истории заказов и контактными данными. Для ускорения обработки запросов реализовано автодополнение email при создании заявок, основанное на кэшировании часто запрашиваемых значений.

Навигационная структура модуля включает адаптивное боковое меню с разделами управления сотрудниками, товарами, заказами и двигателями, а также верхнюю панель (шапку сайта) с функционалом переключения темы и выхода из аккаунта, рисунок 33.

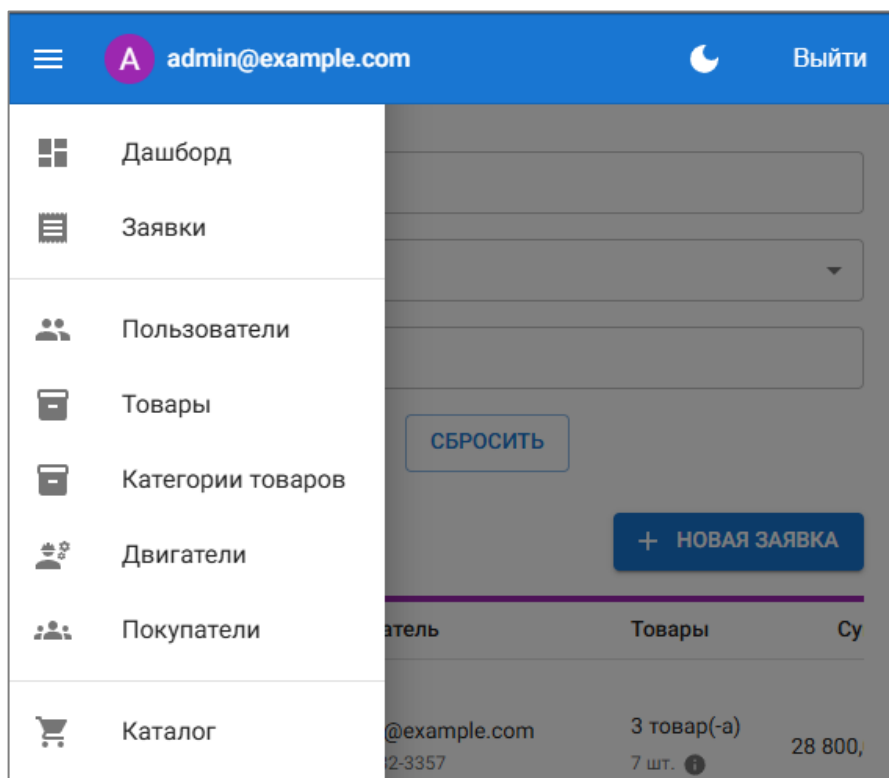


Рисунок 33 – Компоненты бокового меню и шапки сайта

Основу взаимодействия с данными составляют динамические таблицы, поддерживающие пагинацию, сортировку по столбцам и фильтрацию через универсальный компонент таблицы со встроенными фильтрами (рисунок 34), пример таблицы представлен на рисунке 35. Такие страницы реализованы для модулей управления каждой сущностью.

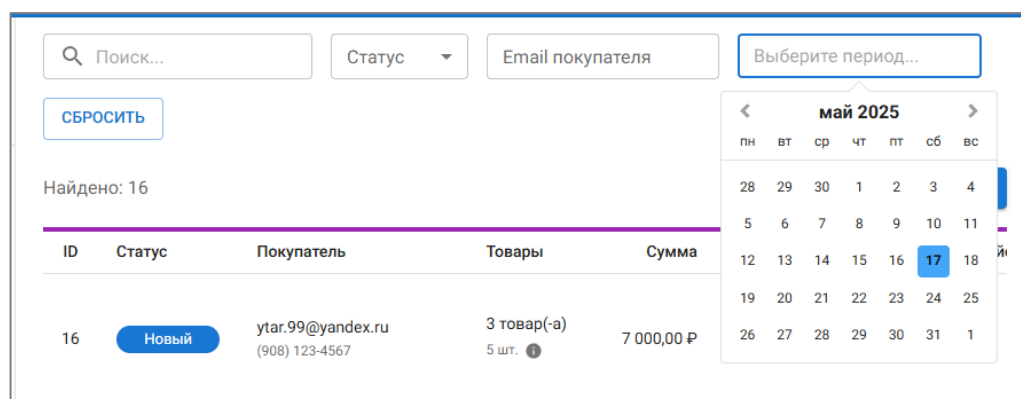


Рисунок 34 – Панель фильтров на странице обработки заявок

Найдено: 2

+
НОВАЯ ЗАЯВКА

| ID | Статус   | Покупатель                            | Товары                 | Сумма       | Создан                  | Изменен                 | Действия                                                                                                                                                                |
|----|----------|---------------------------------------|------------------------|-------------|-------------------------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15 | Завершён | client1@example.com<br>(790) 932-3357 | 3 товар(-а)<br>7 шт. ⓘ | 28 800,00 Р | 17.05.2025,<br>12:28:04 | 17.05.2025,<br>12:28:04 |   |
| 14 | Завершён | client1@example.com<br>(790) 932-3357 | 1 товар(-а)<br>1 шт. ⓘ | 200,00 Р    | 17.05.2025,<br>12:28:04 | 17.05.2025,<br>12:28:04 |   |

Записей на странице: 10
1-2 из 2

Рисунок 35 – Пример с таблицей заявок

Для редактирования сущностей реализованы CRUD-страницы (создание, редактирование и просмотр), где формы ввода интегрированы с правилами валидации (например, для проверки артикулов и цен). Критические операции, такие как удаление записей, сопровождаются диалоговым окном подтверждения для минимизации ошибок. Реализацию страницы редактирования можно увидеть на рисунке 36.

### Редактирование пользователя

Email \*

user15@example.com

Роль

Пользователь

Имя

Андрей

Фамилия

Иванов

Телефон

(795) 068-28-66

Активен

Активен

Изменить пароль

Новый пароль

Повторите пароль

ОТМЕНА

СОХРАНИТЬ

Рисунок 36 – Страница редактирования записи сотрудника

### 3.5 Разработка страниц для взаимодействия с клиентом

Страница каталога реализует двухколоночную структуру, объединяющую навигационное меню категорий и адаптивную сетку товаров, рисунок 37.

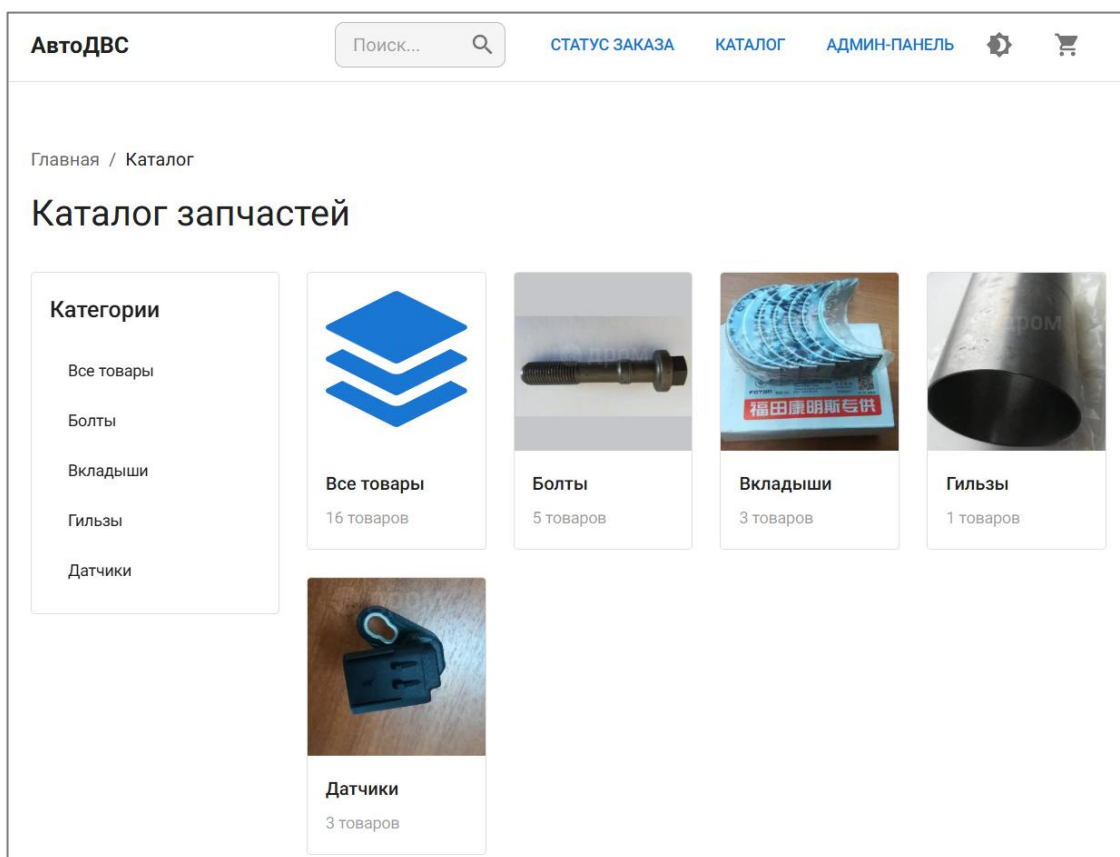


Рисунок 37 – Страница каталога

Левая панель содержит древовидный список категорий с поддержкой многоуровневого вложенного меню, что позволяет пользователю быстро ориентироваться в ассортименте. Основная область использует карточный дизайн с адаптивной вёрсткой, обеспечивающей корректное отображение на мобильных устройствах и больших экранах. Для каждого товара отображается минимизированная информация: изображение, название, артикул и цена, с возможностью перехода к детальному просмотру, рисунок 38.

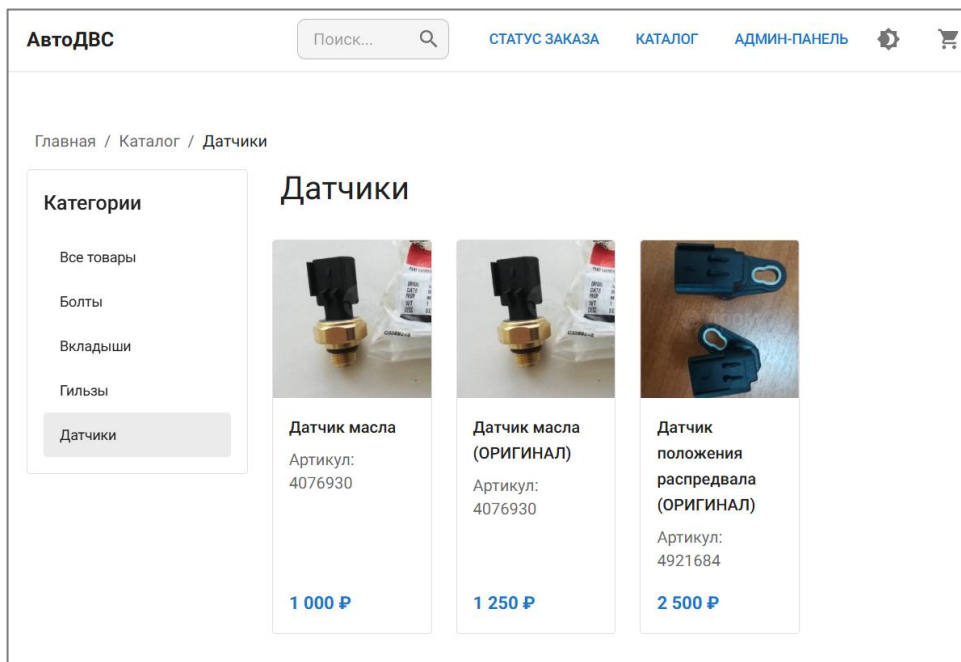


Рисунок 38 – Страница списка товаров в одной из категорий

Механизм бесконечной подгрузки реализован через Intersection Observer API, что снижает нагрузку на сервер при прокрутке [7]. Визуальная обратная связь при загрузке данных обеспечивается скелетонами, имитирующими структуру карточек, рисунок 39.

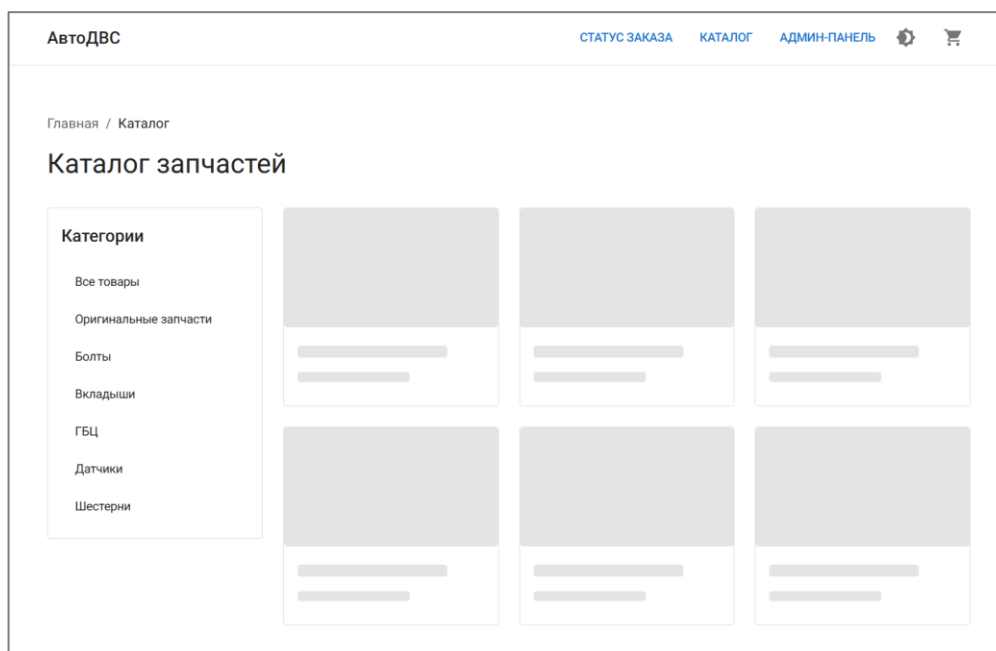


Рисунок 39 – Состояние загрузки контента

Интерфейс детальной информации о товаре разделён на секции: изображение товара, основные характеристики и блок оформления заказа, рисунок 40. Блок метаданных включает динамическое отображение остатков на складе, интерактивный счётчик количества и кнопку добавления в корзину с валидацией допустимых значений [8]. Технические характеристики организованы в табличном виде с фильтрацией по группам параметров. Для сложных товаров реализована система совместимости, отображающая подходящие аксессуары или сопутствующие позиции. Визуальное выделение акционных предложений и гарантийных условий осуществляется через цветовые индикаторы и иконки.

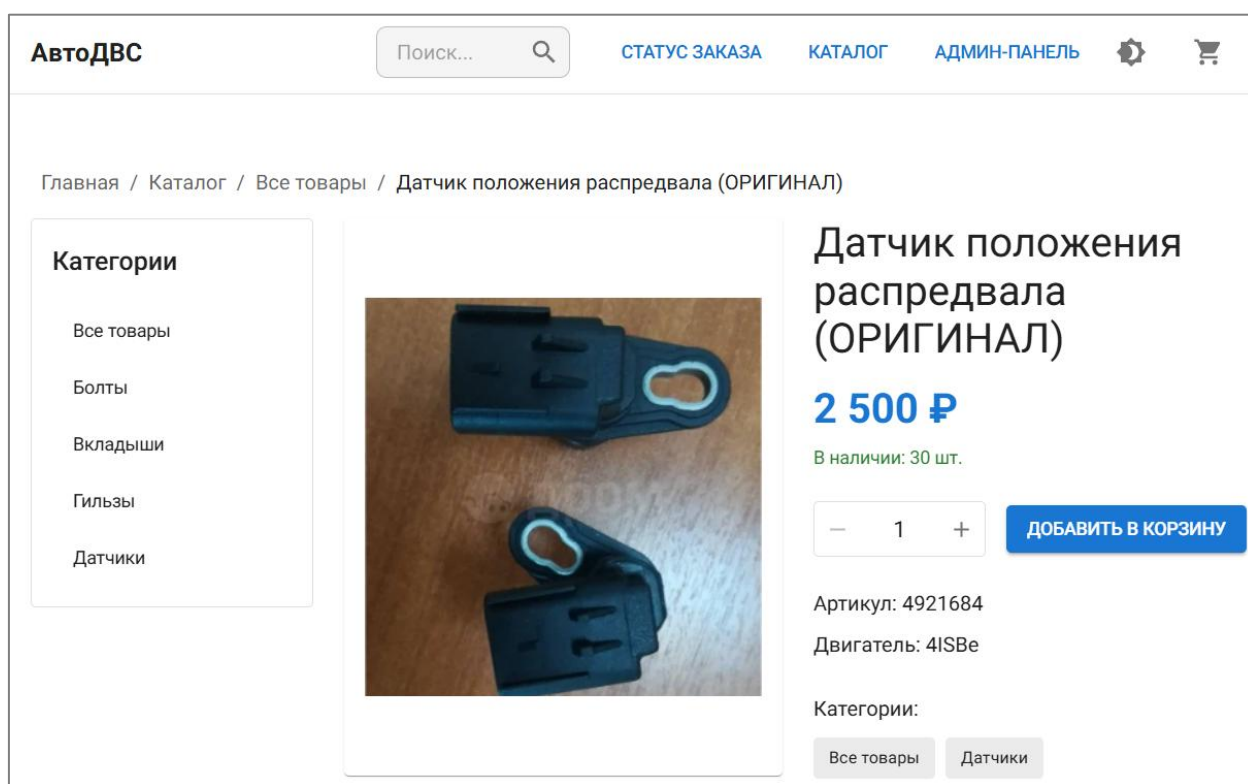


Рисунок 40 – Страница деталей товара

Корзина проектируется как многофункциональный интерфейс, совмещающий список выбранных товаров [9] и форму ввода персональных данных, рисунок 41. Каждая позиция содержит миниатюру товара, редактируемое поле количества с валидацией доступного запаса и интерактивную кнопку удаления. Изменения в количестве мгновенно пересчитывают итоговую сумму через

оптимистичные обновления интерфейса. Блок оформления заказа включает многоуровневую валидацию полей: синтаксический контроль email, проверку формата телефона и обязательность заполнения ключевых реквизитов.

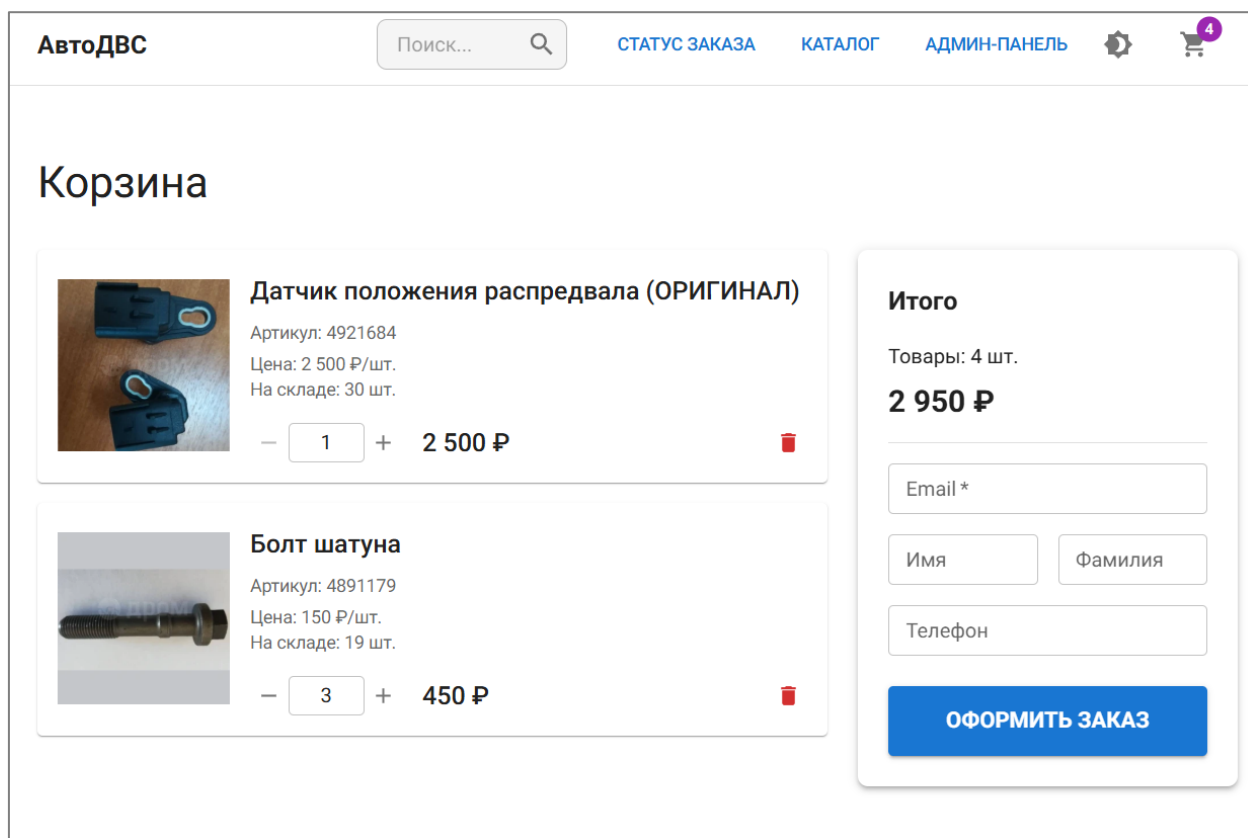


Рисунок 41 – Интерфейс корзины

Интерфейс проверки статуса заказа объединяет форму поиска (рис. 42) и визуализацию этапов обработки (рис. 43). Поисковый блок использует комбинированную валидацию номера заказа и email, исключая возможность несанкционированного доступа. Графическая индикация отображает текущий статус и состав заявки.

## Текущий статус заявки

Номер заказа  
15

Email  
client@example.com

ПРОВЕРИТЬ СТАТУС

Рисунок 42 – Форма ввода для проверки статуса заявки

## Заявка была успешно создана!

**Заявка #16**

На рассмотрении

Дата создания: 04.06.2025

Состав заявки:

|                                                                                     |                                                             |                               |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------|-------------------------------|
|  | Болт шатуна<br>Артикул: 4891179                             | 3 × 150 ₽<br><b>450 ₽</b>     |
|  | Датчик положения распредвала (ОРИГИНАЛ)<br>Артикул: 4921684 | 1 × 2 500 ₽<br><b>2 500 ₽</b> |

**Итого: 2 950 ₽**

Рисунок 43 – Отслеживание статуса

Интеграция с сервисом уведомлений позволяет пользователю получать email-оповещения о создании заказа и критических изменениях статуса (рисунок 44). Для отправки электронных писем используется выделенный SMTP сервер [10], а также отдельный домен, для которого были настроены необходимые DNS записи [11].

## Заказ #16 создан



АвтоДВС no-reply@avtodvs-test.ru  Сегодня в 22:53  
Я >

## Спасибо за заказ!

Номер вашего заказа: #16

### Детали заказа:

- Болт шатуна × 3 - 450 ₽
- Датчик положения распредвала (ОРИГИНАЛ) × 1 - 2500 ₽

**Итого: 2 950 ₽**

Статус заказа можно отслеживать на [странице проверки](#).

Рисунок 44 – Пример электронного письма

Реализованные страницы покрывают все необходимые потребности клиентов и сотрудников по работе со всеми сущностями. Каждый модуль может быть доработан в соответствии с обновляющимися требованиями бизнеса.

## 4 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

Обеспечение безопасности человека в условиях чрезвычайных ситуаций представляет собой важнейшую составляющую современной жизнедеятельности. Безопасность жизнедеятельности как научная дисциплина сосредоточена на изучении потенциальных угроз, возникающих в окружающей среде, и закономерностей их воздействия на человека. Основной целью данной области знаний является разработка комплексных мер, направленных на минимизацию рисков для здоровья людей и сохранение экологического баланса при возникновении опасностей природного или антропогенного происхождения [12].

В рамках настоящей главы рассматриваются ключевые аспекты безопасности, экологичности и предотвращения чрезвычайных ситуаций, актуальные для офисных помещений предприятия ООО «АвтоДВС-28». Анализ включает оценку существующих рисков, методов их нейтрализации, а также рекомендации по обеспечению устойчивого функционирования объекта в условиях внешних и внутренних угроз.

### 4.1 Безопасность

Безопасность представляет собой динамическое состояние, при котором потенциальные угрозы для жизни и здоровья человека минимизируются с учётом вероятности их возникновения и ограничения зоны воздействия. Особенность антропогенных опасностей заключается в их латентности до наступления определённых условий и локальном характере влияния, связанном непосредственно с источником риска. Безопасность жизнедеятельности как научно-практическая дисциплина направлена на сохранение здоровья человека и защиту окружающей среды в условиях профессиональной и бытовой деятельности, что подразумевает разработку превентивных мер противодействия угрозам [12].

Производственная среда, формируемая совокупностью природных и техногенных факторов, способна оказывать негативное влияние на работоспособность и здоровье сотрудников. Вредные факторы, такие как физические перегрузки, гипокинезия или нервно-психическое напряжение, провоцируют развитие профессиональных заболеваний, тогда как опасные факторы при определённых условиях могут привести к острым патологиям или летальным исходам. Соблюдение установленных нормативов организации труда, включая эргономические и гигиенические требования, является ключевым условием предотвращения неблагоприятных последствий [13].

В контексте офисных помещений ООО «АвтоДВС-28» основными рисками выступают физиологические и психоэмоциональные перегрузки, связанные с продолжительной работой за персональными электронно-вычислительными машинами (ПЭВМ). Сидячее положение, занимающее до 70 % рабочего времени, создаёт нагрузку на опорно-двигательный аппарат, несмотря на преимущества в стабильности и энергозатратах. Умственная деятельность, сопровождающаяся обработкой больших объёмов информации, повышает риск стрессовых состояний, снижения реактивности организма и развития сердечно-сосудистых нарушений. Гиподинамия, характерная для подобного труда, усугубляет негативное воздействие на тонус мышц и функциональность внутренних органов.

Классификация трудовой деятельности при работе с ПЭВМ предусматривает три группы: А (чтение информации), Б (ввод данных) и В (творческая работа в диалоговом режиме). Основной вид деятельности определяется по преобладанию временных затрат (не менее 50 % смены). Нормирование нагрузки осуществляется через параметры: количество обрабатываемых символов (для групп А и Б) или продолжительность непосредственного взаимодействия с техникой (для группы В).

Регламентированные перерывы, продолжительность которых регламентируется в зависимости от категории труда (таблица 9), позволяют снизить утомляемость за счёт чередования видов активности. Рекомендуется введение 10–15-минутных пауз каждые 45–60 минут для минимизации зрительного напряжения и статической нагрузки.

Таблица 9 – Суммарное время регламентированных перерывов в зависимости от продолжительности работы, вида и категории трудовой деятельности с ПЭВМ

| Категория работы с ПЭВМ | Уровень нагрузки за рабочую смену при видах работ с ПЭВМ |                             |             | Суммарное время регламентированных перерывов, мин. |                      |
|-------------------------|----------------------------------------------------------|-----------------------------|-------------|----------------------------------------------------|----------------------|
|                         | группа А, количество знаков                              | группа Б, количество знаков | группа В, ч | при 8-часовой смене                                | при 12-часовой смене |
| I                       | до 20 000                                                | до 15 000                   | до 2        | 50                                                 | 80                   |
| II                      | до 40 000                                                | до 30 000                   | до 4        | 70                                                 | 110                  |
| III                     | до 60 000                                                | до 40 000                   | до 6        | 90                                                 | 140                  |

Эргономика как наука о взаимодействии человека с производственной средой играет ключевую роль в организации безопасного пространства. Её задача – адаптация рабочих мест к антропометрическим, физиологическим и психологическим особенностям сотрудников. Критерии оценки включают гигиенические (уровень освещённости, шума, параметры микроклимата) и антропометрические показатели (соответствие мебели и оборудования габаритам пользователя). В помещениях с ПЭВМ обязательными являются: площадь не менее 4,5 м<sup>2</sup> на рабочее место, защитное заземление, исключение близости к источникам электромагнитных помех. Системы кондиционирования обеспечивают поддержание оптимальной температуры, влажности и чистоты воздуха, предотвращая сквозняки и застойные явления [13].

Освещение относится к критически важным факторам безопасности. Естественный свет должен поступать преимущественно с северной или северо-восточной стороны, регулируясь с помощью жалюзи. Искусственное освещение организуется через равномерные или комбинированные системы,

исключающие блики на экранах (максимальная освещённость – 300 лк). Коэффициент пульсации не должен превышать 5 %, что достигается регулярной чисткой светильников и заменой ламп. Дополнительные локальные светильники на рабочих столах обеспечивают комфорт при работе с документами.

Организация рабочих мест регламентируется расстоянием между мониторами (не менее 2 м), зонированием перегородками (1,5–2 м) для задач, требующих концентрации, и эргономичной мебелью. Стулья с регулируемыми параметрами (высота, угол наклона спинки, подлокотники) снижают статическое напряжение. Рабочие столы оборудуются подставками для ног, а клавиатура размещается на расстоянии 100–300 мм от края. Экран располагается на расстоянии 600–700 мм от глаз пользователя, с учётом эргономики зрительного восприятия [14].

Графический интерфейс автоматизированной системы учёта заказов ООО «АвтоДВС-28» разработан в соответствии с принципами эргономики ПО. Минималистичный дизайн, логическая структуризация информации, единообразие элементов управления и подсказки для пользователя снижают когнитивную нагрузку. Экранная плотность оптимизирована для восприятия, а цветовые акценты и иконография направляют внимание на приоритетные данные. Интерфейс адаптирован для пользователей разного уровня подготовки, обеспечивая интуитивную навигацию и минимизацию ошибок ввода.

Таким образом, комплексный подход к безопасности на предприятии включает соблюдение нормативов и непрерывное совершенствование условий труда через внедрение эргономических решений и технологических инноваций.

#### **4.2 Экологичность**

Экология как научная дисциплина формирует основу для разработки стратегий защиты окружающей среды, направленных на гармонизацию взаимодействия между антропогенной деятельностью и природными системами. Её ключевая задача заключается в сохранении биоразнообразия, рациональном использовании ресурсов, минимизации негативного воздействия

производственных процессов на экосистемы и здоровье населения, а также в восстановлении нарушенных природных комплексов.

Разработанный программный продукт, будучи цифровым решением, не оказывает прямого вредного влияния на окружающую среду. Однако эксплуатация сопутствующего технического оборудования — персональных компьютеров (ПК) и multifunctional устройств (МФУ) — сопряжена с экологическими рисками. Наиболее значимым из них является необходимость корректной утилизации устройств по окончании их жизненного цикла. Оборудование, непригодное для ремонта, не может быть размещено на полигонах бытовых отходов из-за содержания токсичных компонентов, таких как тяжёлые металлы, галогены и химические соединения, способных нанести ущерб почве, воде и атмосфере.

Согласно законодательству Российской Федерации, утилизация оргтехники осуществляется исключительно лицензированными организациями, специализирующимися на обращении с отходами I–IV классов опасности. Нарушение этих норм, включая несанкционированный вывоз оборудования, влечёт административную ответственность и значительные штрафные санкции для юридических лиц. В ООО «АвтоДВС-28» процесс утилизации регламентирован поэтапно. Первоначально проводится списание техники с баланса предприятия на основании экспертной оценки, подтверждающей её неремонтопригодность и нулевую остаточную стоимость. Далее заключается договор со специализированной компанией в Амурской области, обеспечивающей транспортировку, демонтаж и сортировку компонентов. Чёрные и цветные металлы, драгоценные элементы направляются на переработку для вторичного использования, а высокотоксичные отходы обезвреживаются или размещаются на лицензированных полигонах. Контроль документального сопровождения каждого этапа исключает риски несоблюдения экологических стандартов.

Отдельное внимание уделяется утилизации отработанных осветительных приборов. Люминесцентные и светодиодные лампы, содержащие ртутьсодержащие соединения, передаются аккредитованным организациям для безопасной переработки. Предприятие ведёт систематический учёт активных, резервных и выведенных из эксплуатации светильников, что позволяет оперативно реагировать на необходимость их замены и минимизировать сроки хранения опасных отходов.

Экологическая политика ООО «АвтоДВС-28» базируется на строгом соблюдении законодательных требований, внедрении ресурсосберегающих технологий и ответственном подходе к управлению отходами. Это обеспечивает снижение антропогенной нагрузки на окружающую среду и формирование корпоративной культуры, ориентированной на устойчивое развитие.

#### **4.3 Чрезвычайные ситуации**

Чрезвычайная ситуация (ЧС) определяется как совокупность событий, возникающих внезапно и создающих угрозу жизни, здоровью людей, экологическому балансу или материальным активам вследствие аварий, природных катастроф, техногенных инцидентов или иных деструктивных процессов. Подобные ситуации характеризуются риском человеческих жертв, масштабными экологическими нарушениями, уничтожением инфраструктуры и устойчивым ухудшением условий жизнедеятельности [12].

Для офисных помещений ООО «АвтоДВС-28» наиболее вероятной ЧС является пожар — неконтролируемый процесс горения, сопровождающийся выделением тепловой энергии, токсичных веществ и разрушением материальных объектов. Противопожарная безопасность на предприятии реализуется через комплекс превентивных и оперативных мер, направленных на минимизацию рисков возгорания, локализацию очагов пламени и обеспечение безопасной эвакуации персонала.

Основными причинами пожаров в офисной среде выступают: нарушение правил эксплуатации электрооборудования, дефекты электропроводки, перегрузка сетей, неосторожное обращение с огнём, а также халатное отношение

сотрудников к установленным нормативам. Наибольшую опасность представляет человеческий фактор, обусловленный пренебрежением к регламентам или недостаточной подготовкой персонала.

Для предотвращения распространения огня в здании предусмотрены противопожарные барьеры — стены, перегородки и двери с повышенной огнестойкостью, ограничивающие перемещение пламени между зонами. Эвакуационные пути, свободные от препятствий и обозначенные световыми указателями, обеспечивают оперативный вывод сотрудников из опасной зоны. Система пожаротушения включает первичные средства (углекислотные огнетушители, пожарные краны) и автоматические установки, активируемые датчиками дыма и тепла. Для ликвидации возгораний применяются вода, порошковые составы и инертные газы, выбор которых зависит от класса пожара и особенностей горящих материалов.

Все сотрудники предприятия обязаны проходить регулярный инструктаж по пожарной безопасности, подтверждая его усвоение подписью в журнале учёта. На рабочих местах размещены схемы эвакуации, а тревожная кнопка позволяет оперативно оповестить экстренные службы. Таким образом, сочетание технических решений, организационных мер и повышения уровня осведомлённости персонала формирует многоуровневую систему защиты, минимизирующую последствия ЧС и обеспечивающую сохранность человеческих ресурсов и активов организации.

## ЗАКЛЮЧЕНИЕ

Задачи и цели данной выпускной квалификационной работы были выполнены в полном объеме. Разработанное веб-приложение для ООО «АвтоДВС-28» представляет собой целостное решение, направленное на устранение ключевых операционных проблем предприятия. Основной акцент сделан на автоматизацию процессов взаимодействия с клиентами, включая формирование заказов, управление товарными остатками и отслеживание статуса заявок. Система объединяет клиентский интерфейс для пользователей, административную панель для сотрудников и централизованную базу данных, что обеспечивает синхронизацию информации на всех этапах работы.

Архитектура платформы построена по модульному принципу, что позволяет гибко адаптировать функционал. Клиентская часть реализует интуитивно понятный интерфейс с динамической загрузкой данных, валидацией ввода и мгновенной обратной связью. Серверный компонент обеспечивает безопасное выполнение транзакций, контроль целостности данных и интеграцию с внешними сервисами.

Система решает проблему ручного управления заказами за счёт автоматизации процессов оформления, включая проверку наличия товаров, расчёт итоговой стоимости и генерацию учётных документов. Интеграция инструментов аналитики позволяет отслеживать ключевые метрики продаж, оптимизировать складские запасы и прогнозировать спрос. Для клиентов реализована возможность самостоятельного отслеживания статуса заказа через защищённый веб-интерфейс, что повышает прозрачность взаимодействия и снижает нагрузку на службу поддержки.

Реализация предложенного решения создаёт основу для цифровизации бизнес-процессов предприятия, сокращения операционных издержек и повышения качества клиентского сервиса, что соответствует стратегическим целям компании в условиях растущей конкуренции на рынке.

## БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

- 1 Агуров П. В. Современные технологии веб-разработки [Электронный ресурс]. СПб. : Питер, 2023. URL: <https://www.litres.ru/book/petr-agurov> (дата обращения: 25.11.2024).
- 2 Документация Next.js [Электронный ресурс]. URL: <https://nextjs.org/docs> (дата обращения: 23.12.2024).
- 3 Документация Prisma ORM [Электронный ресурс]. URL: <https://www.prisma.io/docs> (дата обращения: 24.12.2024).
- 4 Документация React [Электронный ресурс]. URL: <https://react.dev/learn> (дата обращения: 23.12.2024).
- 5 Кузнецов М. И. Тестирование программного обеспечения: методы и инструменты [Электронный ресурс]. М. : ДМК Пресс, 2022. URL: <https://www.litres.ru/book/m-kuznesov> (дата обращения: 11.04.2025).
- 6 Смит Дж. Архитектура программных систем: принципы и паттерны [Электронный ресурс] / Дж. Смит; пер. с англ. М. : Вильямс, 2020. URL: <https://www.litres.ru/book/dzhon-smit> (дата обращения: 26.12.2024).
- 7 Фаулер М. Рефакторинг: улучшение существующего кода [Электронный ресурс] / М. Фаулер; пер. с англ. СПб. : Символ-Плюс, 2019. URL: <https://www.litres.ru/book/martin-fowler> (дата обращения: 13.05.2025).
- 8 Иванов А. С. Разработка интернет-магазинов: от идеи до реализации [Электронный ресурс]. М. : Издательские решения, 2021. URL: <https://www.litres.ru/book/a-ivanov> (дата обращения: 27.12.2024).
- 9 Петрова Е. К. Дизайн пользовательских интерфейсов: карточки товаров [Электронный ресурс]. СПб. : БХВ-Петербург, 2022. URL: <https://www.litres.ru/book/ekaterina-petrova> (дата обращения: 27.12.2024).
- 10 Руководство по SMTP: настройка и управление [Электронный ресурс]. URL: <https://www.mailtrap.io/smtp-guide> (дата обращения: 30.12.2024).
- 11 Настройка DNS для вашего домена [Электронный ресурс]. URL: <https://www.cloudflare.com/learning/dns> (дата обращения: 31.12.2024).

12 Пособие по безопасной работе на персональных компьютерах [Текст] / разраб. В. К. Шумилин. М. : НЦ ЭНАС, 2005. 28 с.

13 Шумилин В. К. ПЭВМ. Защита пользователя [Текст]. М. : Охрана труда и социальное страхование, 2001. 214 с.

14 Кардаш Т. А. Эргономика рабочих мест служащих и инженерно-технических работников, оснащенных ПЭВМ [Текст]: учеб. пособие / Т. А. Кардаш; АмГУ, ИФФ. Благовещенск: Изд-во Амур. гос. ун-та, 2002. 60 с.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Агуров, П. В. Современные технологии веб-разработки [Электронный ресурс] / П. В. Агуров. – СПб. : Питер, 2023. – Режим доступа: <https://www.litres.ru/book/petr-agurov>. – 25.11.2024.
- 2 Брукс, Ф. Мифический человеко-месяц [Электронный ресурс] : пер. с англ. / Ф. Брукс. – М. : Вильямс, 2021. – Режим доступа: <https://www.litres.ru/book/fred-brooks>. – 26.12.2024.
- 3 Васильев, Д. А. JavaScript: продвинутые техники программирования [Электронный ресурс] / Д. А. Васильев. – М. : Эксмо, 2023. – Режим доступа: <https://www.litres.ru/book/dmitriy-vasilev>. – 23.01.2025.
- 4 Галатенко, В. А. Основы информационной безопасности [Электронный ресурс] / В. А. Галатенко. – М. : Интернет-университет информационных технологий, 2021. – Режим доступа: <https://www.intuit.ru/studies/courses/>. – 29.01.2025.
- 5 Гамма, Э. Приемы объектно-ориентированного проектирования [Электронный ресурс] / Э. Гамма, Р. Хелм, Р. Джонсон ; пер. с англ. – СПб. : Питер, 2022. – Режим доступа: <https://www.litres.ru/book/erih-gamma>. – 03.02.2025.
- 6 Документация Material-UI [Электронный ресурс]. – Режим доступа: <https://mui.com/material-ui/getting-started>. – 05.02.2025.
- 7 OWASP Руководство по аутентификации [Электронный ресурс]. – Режим доступа: <https://owasp.org/www-project-web-security>. – 06.03.2025.
- 8 Документация Next.js [Электронный ресурс]. – Режим доступа: <https://nextjs.org/docs>. – 23.12.2024.
- 9 Документация Prisma ORM [Электронный ресурс]. – Режим доступа: <https://www.prisma.io/docs>. – 24.12.2024.
- 10 Документация React [Электронный ресурс]. – Режим доступа: <https://react.dev/learn>. – 23.12.2024.
- 11 ISO/IEC 27001:2022. Информационная безопасность, кибербезопасность и защита приватности – Системы менеджмента информационной

безопасности – Требования [Электронный ресурс]. – Режим доступа: <https://www.iso.org/standard/>. – 13.03.2025.

12 Кент, Б. Экстремальное программирование [Электронный ресурс] : пер. с англ. / Б. Кент. – М. : Лори, 2020. – Режим доступа: <https://www.litres.ru/book/beck-kent>. – 15.01.2025.

13 Кормен, Т. Алгоритмы: построение и анализ [Электронный ресурс] / Т. Кормен, Ч. Лейзерсон ; пер. с англ. – М. : Вильямс, 2023. – Режим доступа: <https://www.litres.ru/book/t-kormen>. – 03.02.2025.

14 Кузнецов, М. И. Тестирование программного обеспечения: методы и инструменты [Электронный ресурс] / М. И. Кузнецов. – М. : ДМК Пресс, 2022. – Режим доступа: <https://www.litres.ru/book/m-kuznesov>. – 11.04.2025.

15 Леоненков, А. В. Проектирование ERP-систем [Электронный ресурс] / А. В. Леоненков. – СПб. : БХВ-Петербург, 2020. – Режим доступа: <https://www.litres.ru/book/a-leonenkov>. – 13.02.2025.

16 Лима, С. Чистая архитектура [Электронный ресурс] : пер. с англ. / С. Лима. – М. : Диалектика, 2021. – Режим доступа: <https://www.litres.ru/book/sandro-lima>. – 15.03.2025.

17 Макконнелл, С. Совершенный код [Электронный ресурс] : пер. с англ. / С. Макконнелл. – М. : Русская редакция, 2019. – Режим доступа: <https://www.litres.ru/book/stiv-makkonnell>. – 16.03.2025.

18 Project Management Institute. Руководство к своду знаний по управлению проектами (PMBOK® Guide) [Электронный ресурс]. – 7-е изд. – США : PMI, 2021. – Режим доступа: <https://www.pmi.org/pmbok-guide-standards>. – 24.03.2025.

19 Шварцмюллер, М. React: продвинутая разработка [Электронный ресурс] : пер. с англ. / М. Шварцмюллер. – М. : Издательский дом "Питер", 2023. – Режим доступа: <https://www.litres.ru/book/maks-shvarcmyuller>. – 25.01.2025.

# ПРИЛОЖЕНИЕ А

## Техническое задание

### **1 Введение**

#### **1.1 Наименование программы**

Разрабатываемая программная система носит наименование «Информационная система управления продажами и клиентскими взаимодействиями ООО «АвтоДВС-28»».

#### **1.2 Краткая характеристика области применения**

Программа предназначена для автоматизации процессов продаж, управления товарным ассортиментом и обработки заказов в рамках деятельности ООО «АвтоДВС-28». Система охватывает функционал взаимодействия с клиентами через публичный веб-интерфейс, а также внутренние бизнес-процессы предприятия, связанные с администрированием заказов, анализом данных и поддержанием базы товаров. Область применения системы ограничена операционной деятельностью предприятия, направленной на повышение эффективности обработки заявок и оптимизацию клиентского сервиса.

### **2 Основания для разработки**

Необходимость создания специализированной информационной системы обусловлена потребностью предприятия в индивидуальном решении, учитывающем специфику его бизнес-процессов. Готовые программные продукты конкурентов не обеспечивают требуемой гибкости адаптации под изменяющиеся требования компании, а также влекут дополнительные расходы на лицензирование и долгосрочное обслуживание. Разработка собственной системы позволит оптимизировать затраты, повысить скорость обработки заказов и обеспечить полный контроль над архитектурой и функционалом.

### **3 Назначение разработки**

#### **3.1 Функциональное назначение программы**

Программа предназначена для реализации следующих ключевых функций: предоставление клиентам доступа к каталогу товаров с возможностью фильтрации и сортировки, формирование заявок через корзину, интеграция с CRM-модулем для управления заказами и клиентскими данными, а также обеспечение аналитической отчетности. Система включает реляционную базу данных для хранения информации о товарах, клиентах, сотрудниках и заказах, что обеспечивает целостность и доступность данных.

#### **3.2 Эксплуатационное назначение программы**

Эксплуатация системы направлена на обеспечение бесперебойного взаимодействия между клиентами и сотрудниками предприятия. Программа позволяет сократить время обработки заказов, минимизировать ошибки ручного ввода данных и повысить прозрачность бизнес-процессов. Использование веб-интерфейса обеспечивает доступ к системе с различных устройств, а модульная архитектура упрощает масштабирование функционала в будущем.

### **4 Требования к программе или программному изделию**

#### **4.1 Требования к функциональным характеристикам**

##### **4.1.1 Требования к составу выполняемых функций**

Программа должна обеспечивать возможность просмотра товаров по категориям, применения фильтров (цена, наличие, характеристики), добавления товаров в корзину и оформления заявок. CRM-модуль обязан включать инструменты управления заказами (смена статусов, назначение ответственных), ведение клиентской базы, формирование аналитических отчетов и визуализацию данных через графики.

## Продолжение ПРИЛОЖЕНИЯ А

### 4.1.2 Требования к организации входных и выходных данных

Входные данные включают информацию о товарах (наименование, артикул, цена, описание), клиентах (контактные данные, история заказов) и заказах (статус, сумма, состав). Выходные данные представлены оформленными заявками, аналитическими отчетами (продажи, остатки) и уведомлениями для пользователей. Форматы данных должны соответствовать стандартам JSON для API-взаимодействия и реляционным моделям для базы данных.

### 4.1.3 Требования к временным характеристикам

Система обязана обрабатывать запросы на получение данных каталога в течение не более 2 секунд при средней нагрузке. Операции оформления заказов и обновления статусов в CRM должны выполняться в пределах 3 секунд. Серверная часть должна выдерживать одновременную работу до 500 пользователей без снижения производительности.

## 4.2 Требования к надежности

### 4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы

Программа должна обеспечивать бесперебойную работу 24/7 за счет использования отказоустойчивых серверных решений и резервного копирования данных. Автоматическое логирование ошибок и мониторинг состояния системы обязательны для своевременного выявления сбоев.

### 4.2.2 Время восстановления после отказа

В случае критического сбоя время восстановления работоспособности системы не должно превышать 15 минут. Резервные копии данных должны обновляться ежечасно для минимизации потерь информации.

## Продолжение ПРИЛОЖЕНИЯ А

### 4.2.3 Отказы из-за некорректных действий оператора

Система должна предотвращать выполнение некорректных операций за счет валидации вводимых данных, подтверждения критических действий (удаление записей) через модальные диалоги и ограничения прав доступа сотрудников в зависимости от их ролей.

### 4.3 Условия эксплуатации

#### 4.3.1 Климатические условия эксплуатации

Программа не предъявляет специальных требований к климатическим условиям, так как функционирует в серверной среде (дата-центре) с контролируемыми параметрами температуры и влажности.

#### 4.3.2 Требования к видам обслуживания

Техническое обслуживание системы ограничивается обновлениями программного обеспечения и мониторингом работоспособности серверов. Профилактические работы должны проводиться в нерабочее время предприятия.

#### 4.3.3 Требования к численности и квалификации персонала

Для эксплуатации системы достаточно одного администратора с базовыми навыками работы с веб-серверами и СУБД. Обучение сотрудников работе с CRM-модулем проводится на этапе внедрения.

### 4.4 Требования к составу и параметрам технических средств

Минимальные требования для серверной части: процессор с 4 ядрами, 16 ГБ ОЗУ, SSD-накопитель 256 ГБ, ОС Linux. Клиентская часть доступна через браузеры Chrome, Firefox, Safari последних версий.

### 4.5 Требования к информационной и программной совместимости

Система должна взаимодействовать с внешними API предприятия через протокол HTTPS. Используемые технологии: Next.js, React, Prisma ORM, СУБД SQLite для стенда разработки и MySQL для продакшена.

## Продолжение ПРИЛОЖЕНИЯ А

### 4.6 Требование к маркировке и упаковке

Программа не требует физической маркировки или упаковки, так как распространяется в цифровом формате.

### 4.7 Требования к транспортированию и хранению

Транспортирование и хранение ограничиваются передачей цифровых носителей с резервными копиями системы в защищенном хранилище.

### 4.8 Специальные требования

Обязательна реализация механизма двухфакторной аутентификации для доступа к CRM-модулю и шифрование конфиденциальных данных клиентов.

## 5 Требования к программной документации

Документация должна включать руководство пользователя для сотрудников, техническое описание архитектуры системы и API, а также инструкции по развертыванию и обновлению.

## 6 Технико-экономические показатели

Внедрение системы позволит сократить ежегодные расходы на лицензирование сторонних решений на 40 %, повысить скорость обработки заказов на 30 % и снизить количество ошибок учета на 25 %.

## 7 Стадии и этапы разработки

Разработка должна быть проведена в три стадии:

1. техническое задание;
2. технический (и рабочий) проекты;
3. внедрение.

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

- разработка программы;
- разработка программной документации;

## Продолжение ПРИЛОЖЕНИЯ А

– испытания программы.

На стадии «Внедрение» должен быть выполнен этап разработки «Подготовка и передача программы».

Содержание работ по этапам:

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

- постановка задачи;
- определение и уточнение требований к техническим средствам;
- определение требований к программе;
- определение стадий, этапов и сроков разработки программы и документации на нее;
- согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы. На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

- разработка, согласование и утверждение порядка и методики испытаний;
- проведение приемо-сдаточных испытаний;
- корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах заказчика.

## Продолжение ПРИЛОЖЕНИЯ А

### **8 Порядок контроля и приемки**

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

Ход проведения приемо-сдаточных испытаний заказчик и исполнитель документируют в протоколе испытаний.

На основании протокола испытаний исполнитель совместно с заказчиком подписывают акт приемки-сдачи программы в эксплуатацию.

## ПРИЛОЖЕНИЕ Б

### Сравнение показателей у аналогичных решений

| Критерий                        | WordPress CRM Plugin                                     | Битрикс24                                                    | AmoCRM                                                   | Собственное решение                                          |
|---------------------------------|----------------------------------------------------------|--------------------------------------------------------------|----------------------------------------------------------|--------------------------------------------------------------|
| <b>Интеграция</b>               | Зависит от WordPress, возможны конфликты обновлений      | Гибкая интеграция, но требует технических ресурсов           | Ограничена предустановленными сервисами                  | Полная интеграция с любыми системами компании                |
| <b>Гибкость настроек</b>        | Базовая кастомизация через плагины (ограничена CMS)      | Сложная настройка из-за избытка функций                      | Ограниченная глубина кастомизации                        | Максимальная адаптация под уникальные бизнес-процессы        |
| <b>Стоимость</b>                | Низкая на старте, но платные расширения                  | Высокая стоимость тарифов, скрытые расходы на обучение       | Дорогие тарифы для расширенного функционала              | Разовая разработка + поддержка, без абонентской платы        |
| <b>Функционал</b>               | Базовые возможности, нет сложной аналитики/автоматизации | Базовые и продвинутые функции (избыточна для малого бизнеса) | Акцент на продажи, слабая аналитика                      | Только необходимые функции (оптимизация под задачи компании) |
| <b>Зависимость от платформы</b> | Риски при обновлениях WordPress, ограничения CMS         | Зависит от облака или серверов компании                      | Облачное решение (риски доступности данных)              | Полная независимость (хостинг, архитектура и обновления)     |
| <b>Сложность внедрения</b>      | Простое, но требует знаний WordPress                     | Требуется обучение и адаптация процессов                     | Интуитивный интерфейс, но сложные сценарии автоматизации | Поэтапное внедрение с учетом специфики бизнеса               |
| <b>Масштабируемость</b>         | Ограничена возможностями WordPress                       | Подходит для крупных компаний, но дорого                     | Проблемы с кастомизацией при росте бизнеса               | Горизонтальное и вертикальное масштабирование                |
| <b>Автоматизация</b>            | Базовые сценарии                                         | Сложные рабочие процессы, но требуют программирования        | Средний уровень (ограничена шаблонами)                   | Любые многоуровневые сценарии                                |
| <b>Безопасность</b>             | Зависит от защиты WordPress (высокий риск взломов)       | Высокая, но облачные данные контролируются поставщиком       | Облачное хранение (риски утечек)                         | Полный контроль данных (шифрование, локальные серверы)       |

Риснок Б.1 – Сравнение показателей у аналогичных решений