

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 09.03.04 – Программная инженерия
Направленность (профиль) образовательной программы – Программная инженерия

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой

_____ А.В. Бушманов

«_____» _____ 2025 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка Android-приложения «Продажа запчастей» для ООО «УникОпт»

Исполнитель

студент группы 1105-об _____ Н.В. Логинов
(подпись, дата)

Руководитель

доцент, канд. техн. наук _____ Т.А. Галаган
(подпись, дата)

Консультант

по безопасности и
экологичности

доцент, канд. техн. наук _____ А.Б. Булгаков
(подпись, дата)

Нормоконтроль

инженер кафедры _____ В.Н. Адаменко
(подпись, дата)

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

УТВЕРЖДАЮ

Зав. кафедрой

_____ А.В. Бушманов
подпись И.О. Фамилия

« ____ » _____ 2025 г.

З А Д А Н И Е

К выпускной квалификационной работе студента Логинова Никиту Владимировича

1. Тема выпускной квалификационной работы: Разработка Android-приложения «Продажа запчастей» для ООО «УникОпт»

(утверждена приказом от 14.04.2025 № 980-уч)

2. Срок сдачи студентом законченной работы (проекта): 10.06.2025

3. Исходные данные к выпускной квалификационной работе: предметная область, перечень литературных источников, отчёт по преддипломной практике.

4. Содержание выпускной квалификационной работы: (перечень подлежащих разработке вопросов): анализ деятельности предприятия, проектирование программного обеспечения, разработка программного обеспечения.

5. Перечень материалов приложения: (наличие чертежей, таблиц, графиков, схем, программных продуктов, иллюстративного материала и т.п.): техническое задание, логическая модель данных, проектирование базы данных и интерфейса.

6. Консультанты по выпускной квалификационной работе (с указанием относящихся к ним разделов): консультант по безопасности и экологичности Булгаков А.Б., доцент, канд. техн. наук.

7. Дата выдачи задания: 14.02.2025 г.

Руководитель выпускной квалификационной работы: Галаган Т.А. доцент, канд. техн. наук

(фамилия, имя, отчество, должность, ученая степень, ученое звание)

Задание принял к исполнению (дата): 14.02.2025 г.

(подпись студента)

РЕФЕРАТ

Бакалаврская работа содержит 82 с., 49 рисунков, 49 таблиц, 25 источников, 6 приложений

КЛИЕНТ-СЕРВЕРНАЯ АРХИТЕКТУРА, МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ, ТЕХНИЧЕСКОЕ ЗАДАНИЕ, UML ДИАГРАММЫ, JETPACK COMPOSE, KOTLIN

В данной работе рассматриваются современные мобильные и клиент-серверные технологии, их возможности в сфере продаж автозапчастей для коммерческого транспорта на базе компании ООО «УникОпт».

Цель работы – разработать мобильное приложение «Продажа запчастей» для ООО «УникОпт».

Для достижения цели работы необходимо произвести: анализ рынка, разработать требования к приложению, выбрать архитектуру и выполнить проектирование интерфейса, реализовать и протестировать программный продукт.

В ходе анализа потребностей компании были сформированы требования к приложению и создано техническое задание. Для реализации приложения была выбрана клиент-серверная архитектура. Готовое приложение включает в себя 15 модулей серверной части, 15 модулей клиентской, 22 таблицы базы данных и более 20 экранных форм. Программа была реализована при помощи Django Framework и Jetpack Compose.

СОДЕРЖАНИЕ

Введение	9
1 Теоретические основы изучаемой проблемы	11
1.1 Общая проблематика предметной области	11
1.2 Характеристика предметной области	11
1.3 Анализ организационной структуры предприятия	14
1.4 Анализ ИКТ организации	16
1.5 Обзор существующих аналогов	17
1.6 Цели создания программного обеспечения и его функции	23
2 Этапы проектирования программного обеспечения	25
2.1 Функции программного продукта	25
2.2 Логическая модель данных	28
2.3 Архитектурный проект	30
2.4 Проектирование программного обеспечения	32
2.5 Проектирование интерфейса	41
2.6 Проектирование базы данных	43
2.6.1 Инфологическое проектирование	43
2.6.2 Логическое проектирование	48
2.6.3 Физическое проектирование	49
2.7 Выбор и обоснование средств реализации	49
3 Описание разработанного программного обеспечения	52
3.1 Общие сведения о программном обеспечении	52
3.2 Функциональное назначение	52
3.3 Логическая структура	53
3.4 Условия применения	62
3.5 Настройка программы	63
3.6 Обращение программе	64
3.7 Основные экранные формы приложения	64

4 Безопасность и экологичность	68
4.1 Безопасность	68
4.1.1 Требования к мобильным устройствам	69
4.1.2 Требования к организации рабочего пространства	70
4.1.3 Требования к освещению на рабочих местах	71
4.1.4 Требования к микроклимату на рабочих местах	72
4.1.5 Требования к организации и оборудованию рабочих мест с ПК	73
4.1.6 Нормирование работы за ПК и перерывов	74
4.1.7 Требования к графическому интерфейсу приложения	75
4.2 Экологичность	76
4.3 Чрезвычайные ситуации	77
Заключение	78
Библиографические ссылки	79
Библиографический список	81
Приложение А	83
Приложение Б	93
Приложение В	88
Приложение Г	108
Приложение Д	120
Приложение Е	127

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей бакалаврской работе использованы ссылки на следующие стандарты:

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

ГОСТ 19.102-77. Стадии разработки.

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

ГОСТ 19.004-80. ЕСПД. Термины и определения.

ГОСТ Р 50571.5.54 Электроустановки низковольтные. Выбор и монтаж электрооборудования. Заземляющие устройства, защитные проводники и защитные проводники уравнивания потенциалов

ГОСТ Р 50948-2001 Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности

ГОСТ Р 52872-2019 Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Приложения для стационарных и мобильных устройств, иные пользовательские интерфейсы. Требования доступности для людей с инвалидностью и других лиц с ограничениями жизнедеятельности

Постановление Правительства РФ от 16.09.2020 N 1479 «Об утверждении правил противопожарного режима в Российской Федерации»

СанПиН 2.1.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания

СП 52.13330.2016 Естественное и искусственное освещение

Федеральный закон «Об отходах производства и потребления» от 24.06.1998 N 89-ФЗ

Федеральный закон «Технический регламент о требованиях пожарной безопасности» от 22.07.2008 N 123-ФЗ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

В настоящей бакалаврской работе применяют следующие термины с соответствующими определениями:

URL (Uniform Resource Locator) – адрес ресурса в интернете, указывающий на его местоположение и способ доступа.

JSON (JavaScript Object Notation) – текстовый формат обмена данными, основанный на синтаксисе JavaScript.

HTTP (HyperText Transfer Protocol) – протокол передачи данных в интернете, основа для обмена веб-страницами.

HTTPS (HyperText Transfer Protocol Secure) – защищённая версия HTTP с шифрованием данных (SSL/TLS).

API (Application Programming Interface) – набор правил и инструментов для взаимодействия программ и клиент-серверных систем.

CRM (Customer Relationship Management) – система управления взаимоотношениями с клиентами.

REST API (Representational State Transfer API) – архитектурный стиль API, использующий HTTP-запросы для обмена данными.

ЭДО (Электронный документооборот) – система обмена юридически значимыми документами в цифровом формате.

MVVM (Model-View-ViewModel) – архитектурный шаблон разработки ПО, разделяющий логику, интерфейс и данные.

1С:Предприятие – платформа для автоматизации бизнес-процессов и учёта, популярная в России.

Android – это мобильная операционная система на базе Linux, предназначенная для смартфонов, планшетов, телевизоров, умных часов и других устройств.

B2B – модель взаимодействия, при которой бизнес продает товары или услуги другим компаниям, а не конечным потребителям.

WebView – это системный компонент в Android, который позволяет встраивать веб-контент в приложение без использования браузера.

СБИС – это российская цифровая платформа для бизнеса, госучреждений и ИП, разработанная компанией «Тензор». Объединяет инструменты для документооборота, отчетности, электронной подписи, работы с госорганами и бизнес-аналитики.

ВИН – это уникальный идентификационный номер транспортного средства, состоящий из 17 символов.

Model-View-Controller – это архитектурный паттерн для разделения приложения на три компонента: логику, интерфейс и данные.

СУБД (Система управления базой данных) – это программное обеспечение для создания, хранения и управления базами данных.

ВВЕДЕНИЕ

В современном мире мобильные технологии играют ключевую роль в развитии бизнеса, предоставляя компаниям новые возможности для взаимодействия с клиентами, оптимизации процессов и увеличения продаж. Одной из таких возможностей является создание мобильных приложений, которые позволяют компаниям выходить на новый уровень обслуживания, обеспечивая удобство, скорость и доступность для пользователей. В условиях растущей конкуренции на рынке, внедрение мобильных решений становится не просто преимуществом, а необходимостью для успешного развития бизнеса.

Компания ООО «УникОпт», занимающаяся оптовой и розничной продажей запчастей, сталкивается с необходимостью модернизации своих каналов взаимодействия с клиентами. В настоящее время основными инструментами продаж являются традиционные методы, такие как телефонные звонки и электронная почта и физические визиты в офис. Однако такие подходы уже не соответствуют современным требованиям рынка, где клиенты ожидают быстрого доступа к информации, удобного поиска товаров и возможности совершать покупки в любое время и в любом месте. Это делает разработку мобильного приложения для Android актуальной и востребованной задачей.

По данным статистики, более 70 % пользователей интернета совершают покупки через мобильные устройства. При этом рынок запчастей для автомобилей и техники также переживает цифровую трансформацию, где ключевыми факторами успеха становятся удобство поиска товаров, прозрачность цен и оперативность доставки. Однако многие компании, включая ООО «УникОпт», до сих пор не используют весь потенциал мобильных технологий, что создает предпосылки для разработки специализированного приложения.

Для разработки необходимо провести анализ потребностей клиентов компании, изучить функциональных возможностей конкурентов, а также технические требова-

ния к приложению. Важно учитывать, что приложение должно быть интуитивно понятным, обеспечивать быстрый доступ к каталогу запчастей, поддерживать функции онлайн-заказа, а также предоставлять клиентам возможность отслеживания статуса заказа.

Актуальность данной работы обусловлена необходимостью повышения конкурентоспособности компании ООО «УникОпт» на рынке продажи запчастей. Мобильное приложение позволит не только упростить процесс взаимодействия с клиентами, но и увеличить объем продаж за счет привлечения новых пользователей и повышения лояльности существующих. Кроме того, приложение станет инструментом для автоматизации бизнес-процессов, что снизит нагрузку на сотрудников и повысит эффективность работы компании.

Практическая значимость работы заключается в создании готового к использованию мобильного приложения, которое будет интегрировано в бизнес-процессы компании. Это позволит организации выйти на новый уровень обслуживания клиентов, увеличить продажи и укрепить свои позиции на рынке.

Целью работы является создание работоспособного Android-приложения, которое обеспечит удобный и быстрый доступ к каталогу товаров, упростит процесс заказа и оплаты, а также повысит уровень удовлетворенности клиентов компании. Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ рынка и потребностей клиентов ООО «УникОпт»;
- разработать функциональные требования к приложению;
- осуществить выбор архитектуры приложения и спроектировать пользовательский интерфейс;
- реализовать функционал приложения, включая работу с каталогом товаров, корзину и выбор способа оплаты и доставки;
- протестировать приложение.

Разработка мобильного приложения для ООО «УникОпт» является важным шагом в развитии бизнеса и повышении его конкурентоспособности.

1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИЗУЧАЕМОЙ ПРОБЛЕМЫ

1.1 Общая проблематика предметной области

Разработка мобильного Android-приложения для продажи запчастей в условиях современного рынка требует решения комплекса задач, связанных с автоматизацией бизнес-процессов и повышением качества обслуживания клиентов. Ключевая проблематика заключается в отсутствии у компании ООО «УникОпт» удобного цифрового инструмента, позволяющего клиентам самостоятельно формировать заказы, отслеживать их статус и получать актуальную информацию о товарах. Это создает избыточную нагрузку на сотрудников, замедляет процесс продаж и снижает удовлетворенность клиентов.

Основные понятия в данной предметной области включают мобильную коммерцию, клиент-ориентированный сервис, интеграцию с CRM-системами и оптимизацию бизнес-процессов. Круг решаемых задач охватывает создание интуитивно понятного интерфейса, разработку функционала для работы с каталогом товаров, реализацию системы заказов и оплаты, а также обеспечение синхронизации данных с внутренними системами компании. Особое внимание уделяется вопросам безопасности персональных данных и бесперебойной работе приложения при высокой нагрузке.

Проблема усугубляется растущими требованиями клиентов к скорости и удобству обслуживания, а также необходимостью для компании сохранять конкурентоспособность в условиях цифровизации рынка автозапчастей. Решение этих задач требует комплексного подхода, сочетающего современные технологии мобильной разработки с глубоким пониманием специфики бизнес-процессов в данной отрасли.

1.2 Характеристика предметной области

Разработка мобильного приложения для продажи автозапчастей представляет собой комплексную задачу на стыке электронной коммерции, корпоративных информационных систем и клиентского сервиса. Предметная область охватывает специфику бизнес-процессов в сфере торговли автомобильными комплектующими для

коммерческого транспорта, особенности мобильной разработки для операционной системы Android, а также требования к интеграции с существующей информационной структурой предприятия.

Методы исследования в этой области могут включать анализ существующих приложений на рынке, опросы и интервью с потенциальными пользователями, а также тестирование интерфейса и пользовательского опыта. Все эти элементы в совокупности создают обширные возможности для внедрения новых решений и улучшения качества обслуживания в сфере продажи запчастей через мобильные приложения.

Продажа запчастей для коммерческого транспорта представляет собой специфический сегмент рынка с ярко выраженной B2B-ориентацией. Основными клиентами в этом бизнесе выступают транспортные компании и индивидуальные предприниматели, владеющие автопарками различного масштаба, для которых принципиально важна возможность безналичных расчетов с официальным документооборотом. При этом существует и менее значительный, но перспективный сегмент частных клиентов – владельцев единичных транспортных средств и небольших станций технического обслуживания, готовых оплачивать запчасти самостоятельно.

Сложившаяся практика работы с крупными корпоративными клиентами предполагает многоэтапный процесс обработки заявок через электронную почту с обязательным участием менеджера на всех этапах – от консультации до выставления счета. Такой подход, обеспечивая индивидуальное обслуживание ключевых клиентов, создает существенную нагрузку на персонал и ограничивает возможности компании по привлечению и качественному обслуживанию мелких заказчиков.

В связи с таким подходом, возникают несколько проблем и возможностей для улучшения процесса, например создание мобильного приложения, которое поможет клиентам формировать заказы без общения с менеджерами.

Ключевыми аспектами предметной области являются особенности рынка автомобильных запчастей с его сложной системой классификации деталей, зависимостью от моделей автомобилей и необходимостью оперативного обновления ассортимента.

Специфика работы с автозапчастями требует учета множества параметров товаров таких как: марка автомобиля, модель, год выпуска, совместимость деталей, артикул, правильное наименование запчасти, производитель и качество, что существенно усложняет структуру каталога по сравнению с другими товарными категориями.

Также немаловажной частью рассматриваемой предметной области является коммерческое мобильное приложение, которое должно соответствовать не только общепринятым стандартам разработки, но и специфике предметной области. Программа должна иметь удобный и интуитивно понятный мобильный интерфейс, систему поиска товаров, каталог, корзину, список заказов и личный кабинет пользователя. Все эти элементы должны работать как единая система, обеспечивая бесперебойный процесс от поиска нужной детали до ее получения клиентом, что требует особого подхода к архитектуре приложения и выбору технологических решений. При этом важно соблюсти баланс между функциональностью приложения и простотой его использования, что является одной из основных сложностей в данной предметной области. Также не стоит забывать о вопросах безопасности при обработке персональных данных клиентов, которые будут собираться и обрабатываться в разрабатываемом программном обеспечении.

Интеграция с существующей CRM-системой и другими корпоративными информационными системами компании является необходимостью в контексте предметной области, которая позволит дополнительно оптимизировать бизнес-процессы и ускорить обработку заказов.

Полученные результаты анализа предметной области служат основой для последующих этапов проектирования архитектуры приложения, выбора технологического стека и разработки детальных технических решений. Выявленные особенности и требования будут учитываться при создании всех компонентов системы – от пользовательского интерфейса до серверной части приложения.

1.3 Анализ организационной структуры предприятия

Компания ООО «УникОпт» была основана в 2007 году как оптово-розничный поставщик бытовых электротоваров. За годы работы организация значительно расширила ассортимент и географию присутствия, превратившись в многопрофильную торговую сеть с диверсифицированными направлениями деятельности.

Основу бизнеса составляет оптовая и розничная торговля широким спектром товаров: от бытовой электроники и садового инвентаря до смартфонов и запчастей для коммерческого транспорта. Особое место в деятельности компании занимает снабжение промышленных предприятий, где «УникОпт» выступает надежным поставщиком комплектующих и оборудования. Такой разнообразный подход к ассортиментной политике позволяет компании сохранять стабильность даже в условиях изменчивого рынка, оперативно перераспределяя ресурсы между наиболее востребованными направлениями.

Важным конкурентным преимуществом «УникОпт» является развитая розничная сеть, представленная магазинами по всей территории Амурской области. Это обеспечивает компании не только устойчивый сбыт, но и прямую связь с конечными потребителями, позволяя оперативно реагировать на изменения спроса.

Для поддержания высокой эффективности бизнес-процессов в компании внедрена современная информационно технологическая (ИТ) инфраструктура. Все рабочие места сотрудников объединены в защищенную локальную сеть с контролируемым доступом в интернет. Основой автоматизации является адаптированная платформа 1С:Предприятие, которая обеспечивает управление товарными запасами, формирование документов и финансовый учет. Электронный документооборот организован через систему «СБИС», что значительно ускоряет взаимодействие с поставщиками и покупателями.

Благодаря сочетанию диверсифицированного подхода к бизнесу, развитой сбытовой сети и современных технологий управления, ООО «УникОпт» демонстрирует устойчивое развитие и остается надежным партнером как для розничных покупате-

лей, так и для крупных промышленных предприятий региона. В перспективе компания планирует дальнейшее расширение ассортимента и внедрение новых технологий для повышения качества обслуживания клиентов.

Рассмотрим организационную структуру предприятия, показанную на рисунке 1. Предприятие имеет линейно-функциональную структуру управления, обеспечивающую четкое разделение обязанностей и эффективное взаимодействие между подразделениями.

Генеральный директор является высшим руководящим звеном, осуществляющим общее стратегическое управление компанией. В его компетенцию входит определение основных направлений развития бизнеса, утверждение ключевых решений, контроль деятельности всех структурных подразделений и обеспечение их согласованной работы.

Коммерческий директор подчиняется непосредственно генеральному директору и отвечает за развитие бизнеса, маркетинг и увеличение продаж. В его ведении находятся вопросы ценообразования, анализ рыночной ситуации, разработка коммерческой политики компании и управление клиентскими отношениями. Коммерческому директору подчиняются отдел продаж и склад предприятия.

Отдел кадров занимается подбором, обучением и развитием персонала, формированием кадровой политики, а также решением вопросов, связанных с трудовым законодательством. Состоит из двух сотрудников, подчиняющихся непосредственно генеральному директору.

Бухгалтерия обеспечивает финансовую деятельность предприятия, включая ведение учета, составление отчетности, расчет заработной платы, налоговое планирование и контроль за финансовыми потоками.

Отдел информационных технологий играет ключевую роль в цифровизации бизнес-процессов. В его функции входит поддержка и развитие IT-инфраструктуры, внедрение программного обеспечения, защита данных предприятия. Отдел состоит из трех сотрудников, поддерживающих информационно-технологическую инфраструктуру предприятия.

Отдел продаж является основным подразделением, непосредственно взаимодействующим с клиентами. Сотрудники отдела занимаются заключением договоров, консультированием клиентов, формированием заказов и контролем их исполнения. Отдел состоит из шести менеджеров, непосредственно подчиняющихся коммерческому директору.

Склад обеспечивает хранение, учет и отгрузку товаров. В его обязанности входит приемка продукции, контроль ее сохранности, формирование отгрузочных документов. Штат склада состоит из девяти человек, обеспечивающих всю работу склада.

Такая структура позволяет предприятию оперативно реагировать на изменения рынка, обеспечивать высокий уровень обслуживания клиентов и поддерживать эффективность внутренних бизнес-процессов. Каждое подразделение выполняет четко определенные функции, что способствует слаженной работе всей организации.

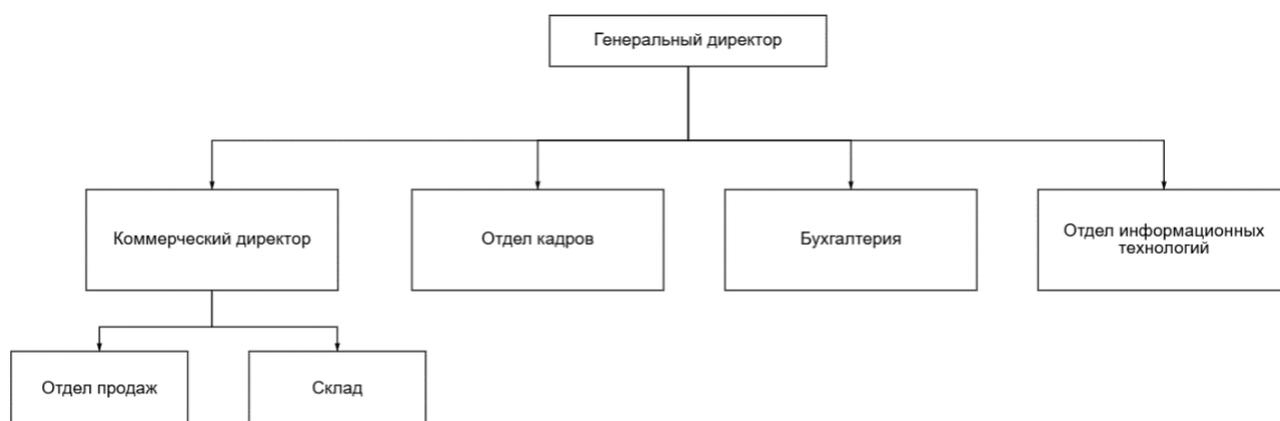


Рисунок 1 – Организационная структура предприятия

1.4 Анализ ИКТ организации

Все сотрудники компании обеспечены персональными компьютерами, объединенными в единую локальную сеть с доступом в интернет. Сетевая инфраструктура построена по иерархической схеме с выделенным серверным оборудованием, обеспечивающим централизованное управление ресурсами и данными. Для поддержания стабильной работы сети используются надежные каналы проводной и беспроводной, а безопасность данных обеспечивается комплексом мер, включающим антивирусную защиту и регулярное резервное копирование важной информации.

Основой автоматизации бизнес-процессов является адаптированная под специфику компании платформа 1С:Предприятие, которая выполняет ключевые функции учета товаров, автоматизированного формирования заказов, выставления счетов и создания закрывающих документов. Система обеспечивает комплексное решение задач складского учета, контроля остатков с закреплением товара по складским ячейкам и штрихкодированием. Благодаря глубокой интеграции с другими корпоративными приложениями такими как CRM Bitrix24 и системами электронного документооборота, 1С автоматизировать основные бизнес-процессы и исключить дублирование информации, что значительно повышает эффективность работы сотрудников.

Информационная безопасность компании обеспечивается за счет строго разделения прав доступа к программному обеспечению используемому в компании. Личные кабинеты сотрудников в программе 1С:Предприятие и Битрикс 24 защищены паролями, известными только самим сотрудникам. Все персональные компьютеры закреплены за определённым сотрудником и так же защищены паролем, что исключает неправомерный доступ к информации. Серверное оборудование защищено как физически (отдельная закрывающаяся комната), так и информационно – системой резервного копирования данных.

Электронный документооборот (ЭДО) организован на базе системы «СБИС» и других специализированных платформ, что позволяет оперативно обмениваться юридически значимыми документами с контрагентами и государственными органами. Все документы подписываются квалифицированной электронной подписью, что обеспечивает их юридическую силу и соответствие требованиям законодательства.

Одним из немногих неавтоматизированных процессов компании является формирование заказа клиента. Текущая инфраструктура компании не предполагает автоматического создания заказов по запросу клиентов с проверкой товара на наличие, поэтому все заказы формируются вручную менеджерами компании что негативно влияет на эффективность труда, особенно в дни пиковой нагрузки.

Существующая информационная инфраструктура организации представляет собой хорошо сбалансированную систему, которая решает основные бизнес-задачи

компании, обеспечивая при этом безопасность данных. Инфраструктура обладает большим потенциалом для дальнейшего масштабирования и расширения в соответствии с растущими потребностями компании, что дает возможность внедрения новых решений призванных повысить эффективность деятельности и уровень обслуживания клиентов.

1.5 Обзор существующих аналогов

Обзор существующих методов решения является важным этапом при формировании требований и проектированию программного обеспечения, позволяя изучить различные подходы к решению задачи, выделить их преимущества и недостатки, информацию о которых можно использовать при проектировании собственной системы.

В качестве объектов для сравнения были выбраны мобильные приложения по продаже автозапчастей «Автодок», «Exist.ru», «Auto3N» и «Гиперавто». Сравнение будет происходить по таким критериям как:

- сложность регистрации, которая будет зависеть от количества данных, которые необходимо ввести для продолжения работы как зарегистрированный пользователь;
- наличие личного кабинета юридического лица, в котором пользователь может ввести реквизиты своей организации и работать от её имени;
- функция сохранения данных автомобиля пользователя для подбора запчастей;
- возможность отслеживания статуса заказов;
- интерфейс пользователя, его общая оценка с точки зрения эргономики, дизайна и удобства для пользователя;
- учет остатков товаров по нескольким складам;
- возможность быстрого заказа товаров, без ввода адреса доставки, способа оплаты, при котором значительно сокращается время оформления заказа, а остальные данные уточняются менеджерами компании.

Каждое из приложений предлагает уникальный набор функций, направленных на упрощение поиска и покупки запчастей. Рассмотрим ключевые аспекты работы этих приложений и выполним сравнение по вышперечисленным критериям.

Сложность регистрации является первым критерием, с которым сталкивается пользователь. В этом аспекте приложения в большинстве своем демонстрируют схожий подход, для регистрации в таких приложениях как «Exist.ru», «Автодок» и «Auto3N» для нормальной работы с приложением требуется ввести очень много данных, такие как номер телефона, фамилия, имя и отчество клиента, адрес электронной почты. В некоторых приложениях так же требуется ввести адрес или выбрать офис обслуживания, что значительно затягивает процесс регистрации и может отпугнуть клиента от дальнейшей работы с приложением. «Гиперавто» выделяется менее детализированной регистрацией, включая ввод ФИО и номера телефона, но это также может являться избыточным при первом знакомстве с приложением.

Наличие личного кабинета юридического лица – важный критерий для профессиональных клиентов, таких как автосервисы и магазины запчастей. Здесь лидером является «Auto3N», который предоставляет функционал для юридических лиц, включая возможность выставление счетов. Однако личный кабинет юридического лица не может быть привязан к личному кабинету физического, что ограничивает функционал. Остальные приложения такого функционала не имеют, и работа с юридическими лицами ограничивается контактами через электронную почту или по телефону.

Функция сохранения данных автомобиля значительно упрощает процесс подбора запчастей, особенно для пользователей, которые регулярно заказывают детали для одной и той же машины. Почти все приложения за исключением «Гиперавто» и «Auto3N» поддерживают эту функцию. «Автодок» и «Exist.ru» позволяют сохранять несколько автомобилей, что полезно для владельцев нескольких транспортных средств или автосервисов.

Возможность отслеживания статуса заказов – ключевая функция для пользователей, которые хотят быть в курсе состояния своих покупок. Все рассматриваемые приложения имеют такую функцию, после оформления заказа он появляется в личном кабинете клиента.

Интерфейс пользователя играет решающую роль в удобстве работы с приложением. «Auto3N» выделяется продуманным дизайном, где основные функции легко доступны, а навигация интуитивно понятна. Работа с каталогом запчастей незначительно усложнена. Дизайн графического интерфейса приложения «Автодок» не является интуитивно понятным, работа с каталогом запчастей сильно усложнена, функция поиска не является интуитивно понятной. Для нормальной работы с приложением требуется потратить значительное время для ознакомления. «Exist.ru» в этом сравнении проигрывает из-за устаревшего дизайна и не всегда логичного расположения элементов. «Гиперавто» отличается удобным интерфейсом, хорошим распределением функционала по вкладкам, но как было выяснено в ходе исследования приложения оно является классическим примером WebView приложения, в котором отсутствует какая-либо логика, а интерфейс представляет собой веб-сайт компании в «обертке» приложения, что негативно влияет на пользовательский опыт так как такие программы обладают увеличенным временем загрузки страниц и плохо оптимизированы.

Учет остатков товаров по нескольким складам критически важен для тех, кто ищет запчасти в разных регионах России. Все приложения обеспечивают такую функциональность включая примерные сроки поставки до адреса клиента.

Возможность быстрого заказа также является важной функцией в приложении поскольку она позволяет клиентам уменьшить время для совершения заказа. Для оформления достаточно сформировать корзину и оформить быстрый заказ введя минимальное количество данных, после чего заказ попадает в обработку менеджеру. Ни одно из рассматриваемых приложений не предоставляет такой функции, то есть пользователям для совершения заказа необходимо ввести множество персональных данных что для некоторых клиентов может быть неудобно.

Проведенный анализ четырех популярных приложений по продаже автозапчастей – «Автодок», «Exist.ru», «Auto3N» и «Гиперавто» – выявил как общие тенденции в их функционале, так и существенные различия, влияющие на пользовательский опыт. На основании этого исследования можно сделать ряд выводов, которые помогут в разработке нового конкурентоспособного приложения в данной сфере.

Прежде всего, стоит отметить, что процесс регистрации в большинстве рассмотренных приложений остается избыточно сложным, требующим ввода значительного объема персональных данных уже на начальном этапе. Это создает барьер для новых пользователей и может негативно сказаться на конверсии. В новом приложении целесообразно реализовать поэтапную регистрацию, когда минимально необходимые данные (номер телефона) запрашиваются сразу, а дополнительная информация (ФИО, данные автомобиля) вносится позже, по мере использования сервиса. Особое внимание следует уделить возможности быстрого старта – например, разрешить поиск запчастей без обязательной регистрации, с предложением зарегистрироваться только при оформлении первого заказа.

Что касается функционала для юридических лиц, то здесь наблюдается явный дефицит качественных решений. Даже в «Auto3N», который предлагает наиболее продвинутые возможности для бизнеса, отсутствует интеграция между личным кабинетом физического и юридического лица. В новом приложении стоит предусмотреть гибридный аккаунт, позволяющий пользователю легко переключаться между персональным и корпоративным режимами работы. Это особенно актуально для владельцев небольших автосервисов и частных мастеров, которые часто совмещают личные и профессиональные потребности в одном аккаунте.

Функция сохранения данных автомобиля, представленная в большинстве приложений, безусловно полезна, но ее реализация часто ограничивается простым хранением марки и модели. В усовершенствованном варианте стоит предусмотреть возможность сохранения полных технических характеристик, VIN-кода, истории заказов и предпочтений. Это позволит не только упростить повторные заказы, но и предлагать персонализированные рекомендации, повышая лояльность пользователей.

В части интерфейса и пользовательского опыта анализ выявил значительный разброс в качестве реализации. Устаревший дизайн «Exist.ru», перегруженность «Автодока» и WebView-решение «Гиперавто» демонстрируют, насколько важно уделить внимание современному, интуитивно понятному интерфейсу. Особое внимание сле-

дует уделить навигации по каталогу запчастей – она должна быть максимально простой и логичной, с эффективной системой фильтрации и поиска. Важно избежать распространенной ошибки, когда сложность каталога, обусловленная большим ассортиментом, переносится на пользовательский интерфейс.

Отслеживание заказов, представленное во всех приложениях, как правило, ограничивается базовым функционалом. В новом решении стоит расширить эту функцию, добавив детализированную информацию о каждом этапе обработки заказа, прогнозируемые сроки доставки, возможность изменения пункта выдачи после отправки, а также интеграцию со службами доставки для онлайн-отслеживания. Это значительно повысит прозрачность процесса и снизит нагрузку на службу поддержки.

Учет остатков на складах реализован достаточно хорошо во всех приложениях, но часто отсутствует информация о динамике цен и прогнозируемых сроках поступления товара. Дополнение этой функции алгоритмами прогнозирования сроков пополнения складов и системой уведомлений о появлении нужной запчасти сделает сервис более удобным для пользователей.

Что касается быстрого заказа, то его отсутствие во всех рассмотренных приложениях представляет собой значительное упущение. В новом решении стоит предусмотреть возможность оформления заказа в минимальное количество шагов, с сохранением платежных данных и адресов доставки для постоянных клиентов. Особенно важно это для мобильного формата, где скорость и простота совершения покупки напрямую влияют на конверсию. Сравнение существующих методов решения показано в таблице 1.

Таблица 1 – Сравнение существующих методов решения

Критерии сравнения	Гиперавто	Автодок	Exist.ru	Auto3N	Предлагаемое решение
Сложность регистрации	Средне сложности	Высокая сложность	Высокая сложность	Высокая сложность	Ввод минимальных данных при регистрации, запрос
Личный кабинет юридического лица	Отсутствует	Отсутствует	Отсутствует	Ограниченная функциональность	Кабинет юридического лица с широкими возможностями

Сохранение данных об автомобилях	Отсутствует	Присутствует	Присутствует	Присутствует	Сохранение данных о нескольких автомобилях
Отслеживание статуса заказов	Присутствует	Присутствует	Присутствует	Присутствует	Расширенная информация о каждом этапе
Учет остатков по нескольким складам	Присутствует	Присутствует	Присутствует	Присутствует	Присутствует
Быстрый заказ	Отсутствует	Отсутствует	Отсутствует	Отсутствует	Присутствует

Проведенный анализ выявил, что ни одно из существующих решений в полной мере не соответствует требованиям ООО «УникОпт», что обусловлено необходимостью сочетания трех ключевых факторов: мобильности клиентского интерфейса, глубиной интеграции с Vitrix24 и учета специфики продаж запчастей. Наиболее перспективным подходом оказалась комбинация мобильного приложения с гибкой серверной частью, позволяющая реализовать все необходимые функции при сохранении возможности адаптации под конкретные бизнес-процессы компании.

1.6 Цели создания программного обеспечения и его функции

Целью создания приложения является автоматизация и оптимизации процессов продаж, улучшение взаимодействие с клиентами и повышение эффективности работы сотрудников. Приложение должно предоставить удобный инструмент для поиска, заказа и отслеживания запчастей, что поможет компании ускорить обработку заказов и повысить уровень обслуживания клиентов.

Ключевой целью создаваемого мобильного приложения является устранение критической проблемы компании, связанной с ухудшением качества сервиса по мере роста клиентской базы ввиду отсутствия удобного инструмента для работы с каталогом запчастей, отслеживания актуальных цен и самостоятельного оформления заказов, что создает дополнительную нагрузку на сотрудников компании. Существующий ручной режим работы не только замедляет обслуживание, но и приводит к перегрузке

персонала, вынужденного выполнять рутинные операции вместо решения стратегических задач. Это особенно критично в условиях роста бизнеса, когда традиционные методы обслуживания перестают быть эффективными.

Разрабатываемое решение призвано кардинально изменить эту ситуацию, перенеся основные процессы взаимодействия с клиентами в цифровую среду. Приложение не просто призвано автоматизировать текущие операции, но и создает принципиально новые возможности для бизнеса – от круглосуточного доступа к услугам до персонализированных предложений на основе истории заказов.

Проведенный анализ позволил сформулировать ключевые цели разработки мобильного приложения для ООО «УникОпт», которые направлены на модернизацию текущих бизнес-процессов компании. Были определены основные функциональные возможности будущего приложения, которые охватывают весь цикл работы с клиентами – от поиска запчастей в каталоге до оформления и отслеживания заказов. Особое внимание уделено функциям, обеспечивающим снижение нагрузки на персонал за счет автоматизации рутинных процессов. Разрабатываемое решение призвано не просто дублировать существующие бизнес-процессы в цифровом формате, но и добавлять новые возможности для бизнеса.

Реализация указанных функций позволит компании достичь стратегических преимуществ: повысить операционную эффективность, улучшить взаимодействие с клиентами и создать основу для дальнейшего цифрового развития бизнеса.

2 ЭТАПЫ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Функции программного продукта

Целью создания приложения является автоматизация и оптимизации процессов продаж, улучшение взаимодействие с клиентами и повышение эффективности работы сотрудников. Приложение должно предоставить удобный инструмент для поиска, заказа и отслеживания запчастей, что поможет компании ускорить обработку заказов и повысить уровень обслуживания клиентов. Для достижения этой цели программное обеспечение должно выполнять следующие функции:

- добавление, удаление товара из корзины, формирование заказов;
- оформление заказов с выбором адреса, способа оплаты и доставки;
- авторизация и регистрация пользователей, личный кабинет клиента;
- каталог товаров с информацией о товаре: наличие товара, его цена, описание;
- сохранение адресов и способов доставки;
- личный кабинет юридического лица;
- система уведомлений о смене статуса заказа клиента;
- сохранение данных о машинах клиента для персональных подборок запчастей;
- административный раздел с функцией редактирования каталога и обновления остатков и цен;
- синхронизация заказов с CRM Bitrix24 компании ООО «УникОпт».

Диаграмма вариантов использования системы «Продажа запчастей», показанная на рисунке 2, представляет собой комплексное отображение функциональных возможностей приложения и взаимодействия между ключевыми участниками системы. В центре архитектуры находится клиент (пользователь), для которого реализованы основные сценарии работы: регистрация и авторизация с аутентификацией, интерактивная работа с каталогом запчастей, полный цикл оформления заказа от формирования корзины до отслеживания статуса доставки, а

также управление персональным гаражом с сохранением данных о транспортных средствах для упрощения будущих покупок.

Менеджер как второй ключевой актер взаимодействует с системой через специализированный функционал: обработка входящих заказов с возможностью изменения статусов и добавления служебных комментариев, синхронизация данных с CRM Vitrix24 в режиме реального времени, включая автоматическую передачу информации о новых заказах и ручное обновление клиентских данных.

Администратор системы обладает расширенными правами управления, включающими конфигурацию параметров работы приложения, мониторинг производительности и устранение инцидентов, настройку автоматического резервного копирования баз данных с возможностью восстановления информации. В зоне ответственности администратора также находится полный цикл управления товарной номенклатурой.

CRM Vitrix24 интегрирована с системой через стандартизированное API и выступает в двух основных ролях: как приемник данных о новых заказах с автоматическим созданием лидов и сделок. Синхронизация осуществляется по защищенному протоколу.

Особенностью архитектуры является модульность – каждый функциональный блок (работа с каталогом, управление заказами, интеграция с CRM) может быть масштабирован или модернизирован независимо от других компонентов системы.

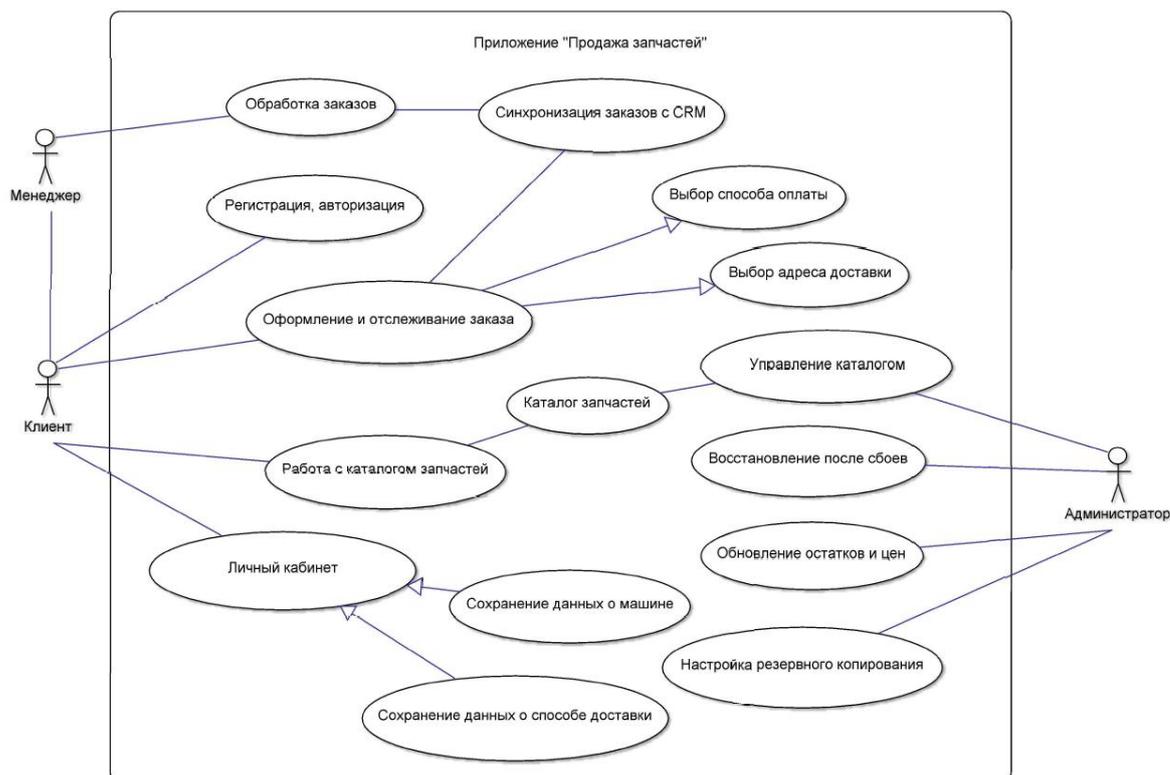


Рисунок 2 – UML-диаграмма вариантов использования

Идентификация и аутентификация пользователей приложения необходима для автоматического закрепления заказов за определенным пользователем, сохранением корзины, адресов, выбранных транспортных компаний и способов доставки. Такой подход улучшает пользовательский опыт, так как пользователю нет необходимости каждый раз при заказе товара вводить адрес доставки. Так же это открывает дополнительные возможности для масштабирования приложения и расширения функционала, например для добавления персональных подборок товаров. Передача персональных данных по сети должна быть реализована с использованием современных методов защиты.

Функции создания, оформления и отслеживания заказов являются базовыми в любом коммерческом приложении для продажи товаров, позволяя пользователю самостоятельно формировать заказы и отслеживать их через личный кабинет пользователя. Страница каталога также является важной частью приложений такого рода предоставляя пользователям инструмент для работы с ассортиментом запчастей

и получении актуальных данных о ценах и наличии товаров. Отличительной особенностью разрабатываемого каталога будет наличие информации о совместимости с различными моделями автомобилей.

Личный кабинет пользователя должен представлять собой страницу, где клиент получает контроль над своим взаимодействием с приложением. Здесь должна быть реализована система управления учетными данными, позволяющая изменять контактную информацию и пароль. Нужно уделить особое внимание разделу истории заказов, где отображается хронологический перечень всех сделанных покупок с детализацией по каждому товару, статусу оплаты и доставки.

Система сохранения данных о транспортных средствах необходима с учетом специфики работы с автозапчастями. Пользователь должен иметь возможность создать профиль каждого своего автомобиля, впоследствии эти данные могут использоваться для подбора запчастей системой или менеджером. При добавлении нового автомобиля в гараж применяется технология постепенного сбора информации – система запрашивает только минимально необходимые данные для начала работы, а остальные данные уточняются в процессе дальнейшей работы.

Корзину товаров необходимо спроектировать как инструмент для формирования заказа. При добавлении товара система автоматически должна проверять его совместимость с указанными пользователем автомобилями, предупреждая о потенциальных проблемах. В процессе оформления заказа клиенту будут доступны различные варианты оплаты и способы доставки.

Интеграция с CRM Vitrix24 должна быть реализована по двунаправленной схеме, обеспечивая синхронизацию данных в реальном времени. При создании нового заказа в мобильном приложении в CRM автоматически необходимо создавать соответствующую сделку с привязкой к карточке клиента.[1]

2.2 Логическая модель данных

Модель IDEF0, показанная на рисунке Б.1 приложения Б демонстрирует архитектуру системы обработки заказов автозапчастей через мобильное приложение, где центральным функциональным блоком выступает само мобильное приложение,

выполняющее преобразование входных данных в результаты работы системы. В левой части схемы расположены основные входные потоки: пользовательские запросы, содержащие параметры поиска и фильтрации; структурированные данные о товарах из каталога; учетные данные клиентов для персонализации сервиса; а также операционные данные, включающие актуальную информацию об остатках на складах и текущих ценах. Эти входные данные поступают в систему через клиентский интерфейс, который служит точкой входа для взаимодействия с пользователями.

Сверху на систему оказывают управляющее воздействие бизнес-правила компании, определяющие логику работы приложения, и политика ценообразования, влияющая на расчет конечной стоимости заказов. В нижней части схемы указаны механизмы реализации: менеджеры компании, выполняющие обработку заказов. Правая часть модели отображает выходные данные системы: сформированные заказы клиентов, которые передаются на дальнейшую обработку в смежные подразделения компании.

Данная модель наглядно показывает, как различные компоненты системы взаимодействуют между собой для достижения основной цели – эффективной обработки заказов автозапчастей. Клиентский интерфейс принимает запросы пользователей и передает их в ядро системы, где они обрабатываются с учетом бизнес-правил и актуальных данных о товарах.

На рисунке Б.2 приложения Б показана диаграмма, отражающая замкнутый цикл взаимодействия между компонентами системы при выполнении типового сценария работы пользователя. Процесс начинается с формирования запросов пользователей через клиентское приложение, которое выступает инициатором всех последующих операций. Приложение передает запрос на сервер приложения, построенный на Django Framework, где происходит его первичная обработка с учетом установленных бизнес-правил компании и актуальной политики ценообразования.

Серверная часть выполняет запросы к базе данных для получения необходимой информации: данных о товарах (включая детальные характеристики и изображения), учетных данных клиентов (профиль, история заказов) и операционных данных

(текущие остатки на складах). База данных обрабатывает запрос и возвращает ответ в формате, оптимизированном для последующей обработки.

Сервер формирует ответ сервера на запрос, который передается обратно в клиентское приложение для отображения пользователю. В случае оформления заказа система выполняет передачу данных о заказе в CRM Bitrix24, обеспечивая синхронизацию с учетной системой компании. Этот процесс происходит в фоновом режиме и включает все необходимые данные для последующей обработки менеджерами компании.

2.3 Архитектурный проект

Представленная на рисунке 3 диаграмма отображает типовой сценарий работы клиента с мобильным приложением «Продажа запчастей» при выполнении поиска товаров в каталоге. Процесс начинается с инициации действия пользователем, который через интерфейс мобильного приложения формирует поисковый запрос с указанием параметров запчасти. Приложение передает этот запрос на сервер, где происходит его валидация и преобразование в структурированный формат.

Серверная часть обрабатывает запрос, формируя соответствующий SQL-запрос к базе данных, который включает условия фильтрации по указанным параметрам. База данных выполняет поиск по полям товарного каталога и возвращает результат, содержащий список соответствующих запросу позиций. Сервер преобразует полученные данные в JSON-формат, оптимизированный для передачи по сети.

Мобильное приложение получает ответ от сервера и выполняет его десериализацию, после чего происходит визуализация результатов в пользовательском интерфейсе.

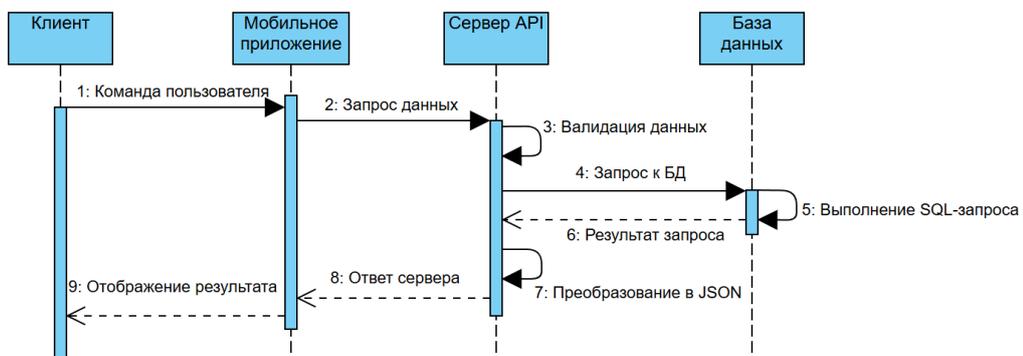


Рисунок 3 – UML-диаграмма последовательности

Представленная схема на рисунке 4 отражает трехуровневую структуру мобильного приложения «Продажа запчастей», реализующую принципы Model-View-ViewModel на клиентской стороне и классическую серверную архитектуру. В клиентском приложении выделены три ключевых компонента: графический интерфейс пользователя (View), отвечающий за визуализацию данных и обработку пользовательских действий, представление модели (ViewModel), которое хранит информацию выводимую в пользовательский интерфейс и преобразующая действия пользователя в команды для модели приложения, и непосредственно модель приложения (Model), выполняющая всю бизнес-логику приложения.[2]

Взаимодействие между клиентом и сервером осуществляется через двунаправленный обмен данными: запросы к серверу, формируемые моделью приложения, содержат параметры поиска, данные аутентификации или информацию о совершаемых действиях (добавление в корзину, оформление заказа). Эти запросы передаются по защищенному HTTPS-соединению в серверное приложение, где модуль обработки запросов выполняет их валидацию, применяет бизнес-правила и формирует SQL-запросы к СУБД. Серверная бизнес-логика включает контроль доступа, проверку наличия товара и другие критически важные операции.

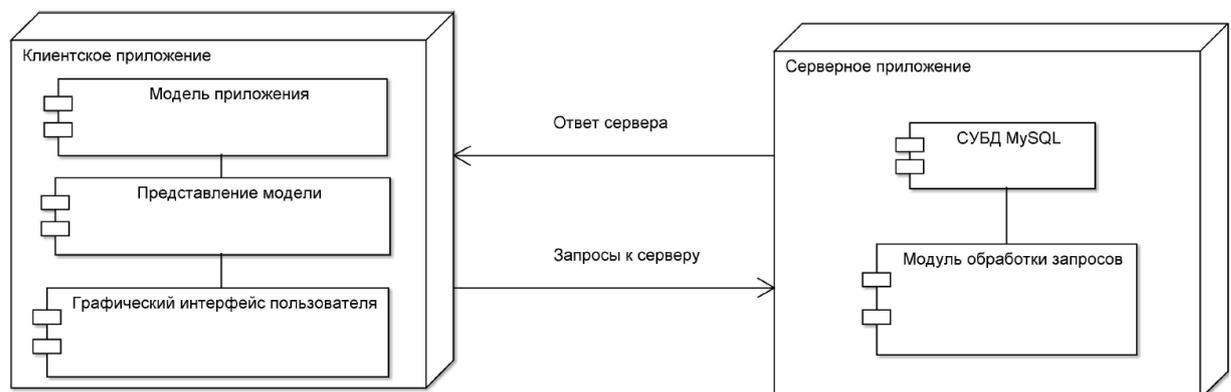


Рисунок 4 – UML-диаграмма компонентов

Ответ сервера в формате JSON содержит запрошенные данные или коды ошибок с поясняющими сообщениями. Полученные данные интерпретируются моделью клиентского приложения и обновляют представление модели, которая

меняет свое состояние и инициирует соответствующее изменение графического интерфейса. СУБД на серверной стороне обеспечивает надежное хранение структурированных данных, поддерживая транзакционность при обработке заказов.

Особенностью архитектуры является четкое разделение ответственности: клиентская часть отвечает исключительно за представление данных и взаимодействие с пользователем, тогда как вся бизнес-логика и работа с постоянным хранилищем вынесена на сервер. Это позволяет обеспечить безопасность критических операций и упрощает обновление логики работы без необходимости модификации клиентских приложений. Обмен данными между уровнями системы стандартизирован и документирован, что обеспечивает возможность независимого развития отдельных компонентов.[3]

2.4 Проектирование программного обеспечения

Наиболее подходящей архитектурой для разрабатываемого программного средства согласно техническому заданию будет являться клиент-серверная архитектура, поскольку она хорошо подходит для реализации приложений с подобным функционалом. Дальнейшую работу по проектированию и разработке можно условно разделить на три этапа:

- проектирование схемы взаимодействия серверной и клиентской части;
- проектирование серверной части, включая базу данных;
- проектирование клиентской части.

От способа и формата взаимодействия серверной и клиентской части будет зависеть дальнейшая архитектура, структура и функциональная составляющая отдельных модулей. Основным методом взаимодействия в клиент-серверных системах является REST API, принцип работы которого заключается в отправке клиентскими приложениями на сервер различных HTTP-запросов и получение на них ответа в JSON или XML формате. JSON формат очень удобен для обмена поскольку большинство языков программирования и фреймворков поддерживают сериализацию в JSON и десериализацию из него. В следствие этого организация взаимодействия клиента и сервера будет заключаться в создании дата-классов в клиентской и серверной частях

приложения, предназначенных для хранения и передачи данных для дальнейшей обработки. Сериализация в JSON формат и извлечение данных из него в дата-объекты будет происходить стандартными методами используемых языков программирования, значительно сокращая время на разработку и отладку функциональных модулей, выполняющих однотипные задачи.[4]

Серверная часть приложения будет представлять собой систему обработки клиентских HTTP-запросов. Разделение по модулям в этом случае лучше всего выполнить согласно перечню решаемых задач.

Первой функциональной группой будут являться работа с каталогом товаров с возможностью просмотра информации о наличии товара, его цене и информации о нем. Для выполнения данных задач было выделено три модуля: «Каталог», «Товар» и «Поиск».

Модуль «Каталог» отвечает за обработку пользовательских запросов к каталогу товаров. В качестве входных данных принимает строку запроса в URL-формате вида «категория/подкатегория/товар», в которой каждый последующий элемент является частью предыдущего представляя иерархическую структуру данных каталога. Также в параметрах строки запроса принимает свойства товаров, по которым выполняется фильтрация и сортировка. При запросе модуль проверяет наличие каталога и возвращает ошибку 404 «Не найдено», в случае если по указанному пути ничего не найдено, и выполняет проверку на активность каталога или товара, то есть если каталог или товар существует, но не активен, то он не возвращается модулем в списке доступных, и при попытке обращения к нему также возвращает ошибку 404. Для ускорения работы модуля может быть применен механизм кэширования.

Выходными данными модуля являются список товаров и подкатегорий, найденных по текущему запросу. В составе ответа должны быть идентификаторы товаров и категорий, используемых при запросе, для формирования дальнейших запросов к серверной части, наименования, описания категорий, сводную информацию о наличии товара без детализации и цене, ссылки на картинки.

Модуль «Товар» отвечает за предоставление детальных данных о запрашиваемом товаре. В качестве входных данных принимает уникальный код товара, передающийся в параметрах строки запроса. При запросе аналогично модулю «Каталог» выполняет проверку на существование и активность товара, если товар не найден или не активен возвращает ошибку 404 «Не найдено». Модуль «Каталог» ассоциирован с модулем «Товар», при обращении к товару через «Каталог» запросы передаются в обработку модулю «Товар».

Выходными данными модуля являются подробные данные о товаре, включая: уникальный код товара, артикул, наименование, описание, производитель, происхождение, доступность товара с ценами и детализацией по складам, совместимость с различными моделями автомобилей.

Модуль «Поиск» выполняет поиск согласно запросу пользователя по каталогу товаров. В качестве входных данных принимает поисковый запрос в параметрах строки запроса. Поиск выполняется в названии товара и его описании, игнорируя регистр. Алгоритм ищет точное соответствие запросу и в качестве выходных данных возвращает список товаров, включая: идентификаторы товаров, их наименования, сводную информацию о наличии товара без детализации и цене, ссылки на картинки.

Для выполнения задач по оформлению заказов, получения их списка, отслеживания статуса, управления заказами и синхронизации с CRM-системой было выделено два модуля: «Заказ» и «CRM интеграция». Доступ к работе с обоими модулями имеют только авторизованные пользователи (прошедшие процесс регистрации).

Модуль «Заказ» отвечает за оформление заказов, предоставление детальной информации и списка заказов. После создания заказа, через модуль «CRM интеграция», с которым связан «Заказ», происходит передача заказа в CRM-систему компании. В качестве входных данных модуль принимает одно из:

- код заказа в параметрах строки запроса для получения детальной информации;
- список товаров с количеством, способ оплаты, адрес доставки, комментарий к заказу и информация о пользователе для оформления заказа;

- запрос с данными пользователя для получения списка заказов пользователя;
- код заказа и наименование действия для отмены заказа.

Выходными данными модуля являются:

– в случае запроса детальной информации о заказе: уникальный код, сумма, статус, дата, статус оплаты и содержание. Содержание заказа включает: уникальный код товара, артикул, наименование, цену покупки и количество;

– в случае успешного оформления заказа возвращает детальную информацию и создает новый заказ, в ином случае ошибку;

– список заказов пользователя, включая уникальный код, сумму, статус, дату и статус оплаты.

Модуль «CRM-интеграция» отвечает за отправку заказов в CRM систему Bitrix24. Входными данными модуля являются состав заказа, адрес доставки, способ оплаты и информация о пользователе. Выходными данными являются код заказа полученный из CRM системы и статус отправки: «Успешно» или «Не успешно».

Модуль «Корзина» выполняет добавление и удаление товаров из корзины, получение списка товаров в корзине пользователя. Автоматически обновляет количество товаров согласно остаткам, то есть если пользователь добавил запчасть в корзину, а через некоторое время товар был куплен, то его количество автоматически изменяется на 0. Доступ к модулю имеют только авторизованные пользователи.

Входными данными модуля являются:

– запрос с кодом товара, количеством и наименованием действия для добавления или удаления товара из корзины;

– запрос с данными пользователя для получения состава корзины пользователя.

Выходными данными модуля в любом случае являются содержимое корзины и результат запроса.

Модуль «Регистрация» отвечает за регистрацию новых пользователей, выполняет проверку переданной информации, такой как электронная почта и номер телефона на существование в базе данных. Сохранение пароля пользователя происходит

при помощи механизма хэширования, во избежание его раскрытия. Входными данными модуля являются имя пользователя, фамилия, электронная почта, номер телефона и пароль. Выходной информацией является сообщение об успешной или не успешной регистрации, запись о новом пользователе в базе данных.

Модуль «Оплата» отвечает за получение доступных способов оплаты. Входными данными модуля являются запросы пользователей. Выходной информацией является список доступных способов оплаты, включая их наименование.

Модуль «Уведомление» предназначен для отправки уведомлений на клиентские устройства. Входными данными модуля являются команды из CRM Bitrix24. Выходной информацией является отправленное уведомление.

Модуль «Авторизация» отвечает за авторизацию пользователей, выполняет проверку введенных учетных данных и в случае успеха аутентифицирует пользователя выдачей уникального токена, который в дальнейшем используется при запросах, требующих авторизации (работа с заказами и корзиной). Входными данными модуля являются идентификатор пользователя (номер телефона) и пароль. Выходными данными пользователя являются токен-аутентификации, в случае успеха или сообщение об ошибке авторизации.

Модуль «Гараж» предоставляет пользователям функционал сохранения моделей автомобилей с их идентификационными данными для дальнейшего подбора запчастей, получения списка автомобилей и управление им. Доступ к модулю имеют только авторизованные пользователи.

Входными данными являются:

- код модели автомобиля, его идентификационные данные и информация о пользователе для сохранения машины в гараже;
- запрос с данными пользователя для получения списка машин;
- код машины в гараже, наименование действия для удаления машины из гаража или изменения информации о ней.

Выходными данными в любом случае являются список машин, включая модели, производителей, идентификационные данные и результат запроса.

Модуль «Адрес» выполняет сохранение адресов доставки пользователя, получения их списка и управления им. Позволяет удалять и редактировать адреса доставки. Доступ к модулю имеют только авторизованные пользователи.

Входными данными являются:

- адрес (регион, город, улица, дом) и уникальный код транспортной компании для сохранения адреса;
- запрос с данными пользователя для получения списка сохраненных адресов;
- код адреса, наименование действия для удаления адреса или его редактирования;
- код адреса для получения детальной информации.

Выходными данными в любом случае являются сохраненные адреса, включая адрес доставки, наименование транспортной компании и результат запроса.

Модуль «Транспортная компания» отвечает за получение доступных транспортных компаний. Входными данными модуля являются запросы пользователей. Выходной информацией является список доступных транспортных компаний включая их наименование и описание.

Модуль «Обработчик URL» является главной связующей частью всех модулей, поскольку именно он отвечает за первичную обработку запросов пользователей и переадресацию их выполнения в соответствующие модули. Входными данными модуля являются запросы пользователей, множество выходных данных представляет из себя совокупность выходной информации модулей, к которым была выполнена переадресация.

Для защиты серверной части от несанкционированного доступа был спроектирован модуль «Ограничение доступа». Все запросы к серверу должны содержать уникальный ключ в заголовке запроса, записанный в базе данных приложения, его проверку выполняет данный модуль. Входной информацией являются запросы пользователей. Выходными данными, в зависимости от результата проверки, являются либо

отказ в доступе, если ключ не указан или не является правильным, либо переадресация запроса в модуль «Обработчик URL», который ассоциирован с описываемым. UML-диаграмма взаимодействия модулей показана на рисунке В.1 приложения В.

Далее необходимо выполнить проектирование модулей клиентской части приложения.

Группа модулей «Работа с сервером» отвечает за выполнение запросов к серверной части приложения и преобразование ответа сервера в пригодный для дальнейшей обработки формат, в его состав входят более мелкие модули каждый из которых выполняет свой перечень задач. Входными данными всех подмодулей являются запросы, снабженные дополнительной информацией, в зависимости от выполняемых задач, выходными данными являются структурированный ответ включая все данные, присланные сервером и код ответа.

Модуль «Работа с заказами» выполняет запросы к серверу, связанные с оформлением заказов, их отслеживанием и отменой. Входными данными модуля являются:

- токен аутентификации пользователя, список товаров с количеством, информация о доставке для оформления заказа;
- токен аутентификации пользователя, команда пользователя, код заказа для просмотра списка заказов или их отмены.

Модуль «Работа с каталогом» выполняет запросы, связанные с каталогом товаров (поиск, детальный просмотр товаров). Входными данными модуля являются:

- путь к товару или категории с параметрами сортировки и фильтрации для навигации по каталогу, выходными данными в этом случае будет список товаров и подкатегорий информацией о них;
- поисковая строка для выполнения поиска по каталогу, выходной информацией в этом случае будет список найденных товаров.

Модуль «Работа с корзиной» выполняет запросы, связанные с корзиной товаров: добавление и удаление, синхронизация остатков. Входными данными модуля являются: список товаров с кодами и количеством, выходной информацией является список товаров в корзине, синхронизированный по остаткам и ценам с сервером.

Модуль «Работа с адресами» выполняет запросы, связанные с сохраненными адресами пользователей. Входными данными модуля являются:

- токен аутентификации пользователя, адрес и код транспортной компании для добавления нового адреса;
- токен аутентификации пользователя, команда пользователя, код адреса для просмотра списка адресов и их изменения.

Выходной информацией модуля является список сохраненных адресов пользователя.

Модуль «Работа с пользователями» выполняет запросы, связанные с регистрацией и авторизацией пользователей. Входными данными модуля являются:

- логин и пароль пользователя, для авторизации, выходной информацией в этом случае является токен аутентификации в случае успешного входа или сообщение об ошибке;
- электронная почта, имя, фамилия, пароль и номер телефона пользователя для выполнения регистрации, выходной информацией в этом случае является сообщение об ошибке или успешной регистрации.

Модуль «Работа с уведомлениями» принимает уведомления для устройства с сервера и выводит их на экран, работает в фоновом режиме, в качестве входных данных принимает информацию с сервера, выходными данными являются уведомления для пользователей.

Модуль «Работа с гаражом» выполняет запросы, связанные с редактированием содержимого гаража пользователя и управления им, входной информацией является токен аутентификации, данные добавляемой или удаляемой машины. Выходной информацией являются информационные сообщения и список машин в гараже пользователя.

Следующим этапом является проектирование группы модулей отвечающих за обработку действий пользователя с интерфейсом системы и обращение к модулям, отвечающим за работу с сервером. Данная группа фактически будет являться компонентом ViewModel архитектуры MVVM, которая была выбрана для проектирования.

Следовательно перечень выполняемых модулем задач будет полностью соответствовать списку функций экранной формы или группе экранных форм. Все действия пользователя (нажатия, ввод данных) будут обрабатываться графическим интерфейсом при помощи стандартных методов операционной системы, поэтому нет необходимости в реализации данного функционала.

Модулем, отвечающим за работу с каталогом запчастей, поиск, добавление товаров в корзину будет являться «Каталог». Входными данными модуля являются команды пользователя и строки запросов, выходной информацией – результаты этих запросов, а именно: списки найденных товаров, детальная информация о запчастях и информационные сообщения. Компонент выполняет все команды поступающие от главного экрана приложения включая его дочерние формы.

Для выполнения задач по авторизации и регистрации был выделен модуль «Регистрация и авторизация». Входными данными модуля являются команды пользователей и информация для регистрации или авторизации, выходной информацией – результаты этих запросов и информационные сообщения. Компонент выполняет все команды поступающие формы входа и регистрации приложения.

Обработка команд по оформлению заказов возложена на модуль «Оформление заказа», он выполняет команды, поступающие от всех экранных форм, связанных с оформлением заказа. Входными данными модуля являются команды пользователя, информация о доставке и способе оплаты, выходной информацией – результаты этих запросов и информационные сообщения.

Взаимодействие с корзиной запчастей выполняется при помощи модуля «Корзина», он выполняет команды, поступающие от формы с корзиной. В функционал формы так же заложен переход к оформлению заказов, в этом случае управление передается компоненту «Оформление заказа». Входными данными модуля являются команды пользователя, выходной информацией – результаты этих запросов и информационные сообщения.

Модуль «Профиль» отвечает за работу с данными пользователя в личном кабинете, редактирование информации. Входными данными модуля являются команды

пользователя, изменяемая информация, выходной информацией – результаты этих запросов, информационные сообщения и данные профиля пользователя.

За просмотр списка заказов и отслеживание их статуса отвечает компонент «Заказ». Входными данными модуля являются команды пользователя, выходной информацией – результаты этих запросов, информационные сообщения и данные по заказам.

Модуль «Адрес» предназначен для работы с сохраненными адресами пользователя, для их просмотра, редактирования и удаления. Входными данными модуля являются команды пользователя, изменяемая информация, выходной информацией – результаты этих запросов, информационные сообщения и список адресов.

Для взаимодействия с виртуальным гаражом пользователя был выделен компонент «Гараж», он выполняет задачи по изменению состояния гаража и отображению его содержимого. Входными данными модуля являются команды пользователя, данные об автомобилях, выходной информацией – результаты этих запросов, список машин в гараже и информационные сообщения.

Модули «Гараж», «Профиль», «Заказ», «Адрес» и «Регистрация и авторизация» выполняют задачи, поступающие из личного кабинета пользователя и его дочерних форм. В зависимости от формы управление передается соответствующему модулю.

Все описанные компоненты вне зависимости от выполняемых задач должны производить верификацию введенных данных и всех команд пользователя, и в случае обнаружения ошибки отправлять пользователю информационные сообщения. Это позволит снять излишнюю нагрузку с сервера и сократить время ответа при неправильно введенных данных. Диаграмма взаимодействия модулей серверной части показана на рисунке В.2 приложения В.

2.5 Проектирование интерфейса

Интерфейс мобильного приложения состоит из более чем двадцати экранных форм, основными из них являются:

– «Каталог» для поиска товаров, просмотра детальной информации и добавления в корзину;

– «Корзина» для просмотра списка добавленных товаров, их удаления и последующего перехода к оформлению заказов;

– «Личный кабинет» для просмотра списка заказов, редактирования личной информации, просмотра и добавления машин в «Гараж».

Главным экраном приложения будет являться каталог товаров, прототип которого показан на рисунке В.3 приложения В. С точки зрения проектирования он выполнен в минималистичном стиле, содержит только ключевую информацию: название товара, артикул, статус наличия и цену. Основными элементами взаимодействия являются: кнопка «Купить», которая выделена цветом для привлечения внимания, кнопки поиска и сортировки в верхней части экрана. В нижней части экрана расположена навигация по основным экранам приложения. С точки зрения UX, интерфейс интуитивно понятен: пользователь сразу видит важные детали товара и может быстро совершить действие.[5]

Следующий экран «Корзина» представляет собой интерфейс, отображающий карточки товаров, добавленных в корзину. Карточка содержит информацию о товаре: название, артикул и модель двигателя, цену и наличие на складе. Это позволяет пользователю быстро убедиться в корректности выбора. Под описанием товара размещены интерактивные элементы управления: кнопка «Удалить» для исключения товара из корзины и счетчик количества, дающий возможность изменить объем заказа. В нижней части экрана расположены кнопки «Оформить заказ» и «Быстрый заказ», для перехода к оформлению заказа двумя различными способами. Рядом с кнопками расположена сумма заказа и сумма НДС (Налога на добавленную стоимость). Дизайн экрана выполнен в минималистичном стиле с акцентом на функциональность. Отсутствие лишних элементов снижает нагрузку на пользователя, а четкая структура помогает пользователю сосредоточиться на совершении целевого действия – оформлении заказа. [6] Экран показан на рисунке В.4 приложения В.

Личный кабинет пользователя в мобильном приложении представляет собой ключевой элемент навигации, сочетающий простоту интерфейса с функциональной

насыщенностью. Центральную часть экрана занимает список опций профиля, включающий «Заказы», «Адреса», «Настройки», «Помощь», «Гараж», «Профиль юридического лица» и «Политика конфиденциальности», что охватывает все основные потребности пользователя в рамках личного кабинета.

UX-составляющая экрана продумана для максимального удобства: основные разделы доступны в один клик, а логичная группировка функций ускоряет выполнение типовых задач. Например, часто используемые разделы, такие как «Заказы» и «Адреса», расположены в верхней части списка, тогда как вспомогательные элементы «Политика конфиденциальности» находятся внизу, что соответствует принципам приоритизации.[5] Экран показан на рисунке В.5 приложения В.

2.6 Проектирование базы данных

2.6.1 Инфологическое проектирование

На этом этапе проектирования будет выполнено инфологическое проектирование базы данных, в рамках которого будут определены сущности и список их атрибутов.[7]

Сущность «Пользователь», описанная в таблице В.1 приложения В, является центральным элементом системы и включает полный набор атрибутов для идентификации и аутентификации. Уникальный числовой код служит первичным ключом. Логин совмещен с электронной почтой, что обеспечивает унифицированный вход в систему. Флаги администратора и активности имеют логический тип данных и определяют уровень доступа. Также сущность имеет атрибут для хранения номера телефона пользователя.

Сущность «Адрес», описанная в таблице В.2 приложения В, структурирована по принципу от общего к частному: регион, город, точный адрес. Точный адрес включает улицу, номер дома и квартиру в едином текстовом поле, что соответствует стандартам почтовой доставки. Все текстовые поля имеют неограниченную длину для корректного отображения сложных адресов.

Сущность «Ключ доступа», описанная в таблице В.3 приложения В, содержит сгенерированные криптостойкие ключи, включающие буквы разного регистра и цифры. Ключи используются для API-аутентификации.

Сущность «Корзина», описанная в таблице В.4 приложения В, представляет собой промежуточный контейнер между каталогом и заказом. Выполняет хранение содержимого корзины пользователя. Количество товара имеет положительное целочисленное значение. Сущность не содержит информации о самом товаре, только его количестве.

Сущность «Заказ», таблицы В.5 приложения В, фиксирует финансовые параметры заказов клиентов. Сумма хранится в формате с двумя знаками после запятой для точного отображения денежных значений. Дата заказа автоматически устанавливается как текущая при создании записи. Статус оплаты логический, если «истина», то заказ оплачен, в ином случае не оплачен. Интеграционные поля с CRM содержат внутренние идентификаторы Vitrix24.

Сущность «Происхождение товара», описанная в таблице В.6 приложения В классифицирует товары по типу происхождения. Включает код и текстовое наименование.

Сущность «Тип цены», таблицы В.7 приложения В определяет категории ценовой политики. Состоит из числового кода и наименования типа цен.

Сущность «Цена на товар», показанная в таблице В.8 приложения В, фиксирует стоимость товаров. Содержит код и числовое значение цены, ограниченное положительными числами.

Сущность «Картинка», таблицы В.9 приложения В, реализует паттерн «один товар – много изображений». Пути хранятся в формате относительных URL от корня сервера. Поле «Главная картинка» определяет изображение для превью в каталоге. Миниатюры генерируются автоматически при загрузке с фиксированным разрешением.

Сущность «Каталог», описанная в таблице В.10 приложения В, содержит технические характеристики товаров. Артикул соответствует внутренней нумерации

производителя. Поле «Активность» позволяет временно скрывать товары без удаления.

Сущность «Производитель», таблица В.11 приложения В, хранит информацию о компаниях-производителях запчастей. Состоит из кода и наименования производителя.

Сущность «Способ оплаты», показанная в таблице В.12 приложения В, определяет варианты оплаты заказов. Включает код и наименование способа.

Сущность «Транспортная компания», таблицы В.13 приложения В описывает службы доставки. Содержит код, наименование транспортной компании и текстовое описание услуг.

Сущность «Статус заказа», описанная в таблице В.14 приложения В, фиксирует этапы обработки заказов. Состоит из кода и наименования статуса заказа.

Сущность «Склад», показанная в таблице В.15 приложения В, описывает склады компании. Включает код, наименование и адрес склада.

Сущность «Юридическое лицо», описанная в таблице В.16 приложения В, содержит реквизиты компаний. Состоит из кода, наименования, ИНН, КПП, адреса, банковских реквизитов, таких как расчетный счет, БИК.

Сущность «Содержание заказа», таблицы В.17 приложения В, детализирует состав заказов. Включает код, цену и количество товара.

Сущность «Остаток товара», таблицы В.18 приложения В, фиксирует наличие товаров на складах. Содержит код и количество товара.

Сущность «Категория товара», описанная в таблице В.19 приложения В, реализует древовидную структуру каталога через поле «Код родителя». Уровень вложенности автоматически рассчитывается при изменении иерархии. Символьный код используется для человекопонятных-ссылок и должен быть уникальным. Активность категории влияет на видимость всех вложенных товаров и подкатегорий.

Сущность «Токен аутентификации» в таблице В.20 приложения В управляет сессиями пользователей. Содержит текстовый токен, дату его создания и идентификатор устройства, на котором выполнен вход.

Сущность «Гараж», описанная в таблице В.21 приложения В, выполняет хранение информации о транспортных средствах пользователя с их уникальными идентификаторами – ВИН-кодами.

Сущность «Производитель машины» в таблице В22 приложения В описывает автомобильные марки, используемые в приложении. Включает код и наименование производителя.

Сущность «Модель машины», описанная в таблице В.23 приложения В, содержит технические спецификации машины в поле «Описание». Наименование модели соответствует официальному обозначению производителя.

Первичные ключи являются неотъемлемой частью любой реляционной базы данных, потому что они позволяют однозначно идентифицировать каждую отдельную запись в базе данных. Каждый экземпляр сущности в проектируемой базе данных должен иметь свой уникальный код. Для удобства разработки базы данных в качестве первичного ключа каждой сущности был создан атрибут целочисленного типа под названием «Код».

Сущность «Пользователь» связана с «Адресом» отношением «один-ко-многим», где один пользователь может иметь несколько адресов доставки. С «Корзиной» установлена связь «один-ко-многим» – каждому пользователю соответствует несколько товаров в корзине. Отношение с «Заказом» также «один-ко-многим», позволяя пользователю иметь несколько заказов.

Сущность «Гараж» связана с «Пользователем» отношением «один-ко-многим», так как пользователь может хранить данные нескольких автомобилей. «Токен аутентификации» имеет связь «один-ко-многим» с «Пользователем», позволяя одному пользователю иметь несколько активных токенов.

«Каталог» товаров связан с «Производителем» отношением «многие-к-одному», где несколько товаров могут относиться к одному производителю. С «Категорией товара» установлена связь «многие-к-одному», позволяя классифицировать товары по категориям. Отношение с «Происхождением товара» имеет вид «многие-к-

одному», то есть каждому типу происхождения может соответствовать несколько товаров, определяя тип производителя для каждого товара.

Сущность «Содержание заказа» связана с «Каталогом» отношением «многие-к-одному», где каждая позиция заказа ссылается на конкретный товар. С «Остатком товара» установлена связь «один-ко-многим», позволяя отслеживать остатки по каждому товару на разных складах.

«Остаток товара» связан со «Складом» отношением «многие-к-одному», фиксируя наличие товаров на конкретных складах. То есть одной записи об остатках товара соответствует один склад и одному складу соответствует несколько записей об остатках. «Цена на товар» имеет три связи: с «Каталогом», «Складом» и «Типом цены», образуя сложную систему ценообразования.

Сущность «Адрес» связана с «Транспортной компанией» отношением «многие-к-одному», определяя доступные способы доставки для каждого региона. «Содержание заказа» имеет связь «многие-к-одному» с «Заказом», формируя состав каждого заказа.

«Заказ» связан с «Способом оплаты» отношением «многие-к-одному», и с «Адресом» аналогичным отношением, фиксируя условия каждой сделки. «Юридическое лицо» имеет связь «один-ко-многим» с «Пользователем», позволяя привязывать корпоративные реквизиты к учетным записям.

Сущность «Картинка» связана с «Каталогом» отношением «многие-к-одному», обеспечивая визуальное представление товаров. «Модель машины» имеет связь «многие-к-одному» с «Производителем машины», «один-ко-многим» с «Гаражом», образуя систему идентификации транспортных средств и привязки их к пользователю.

Дополнительно «Модель машины» связана с «Каталогом» через отношение «Совместимость», для устранения связи «многие-ко-многим». Сущность определяет какие запчасти подходят для конкретных автомобилей. Все связи реализованы через внешние ключи, обеспечивающие целостность данных.

В результате инфологического проектирования базы данных был сформирован и описан список сущностей необходимых для реализации функционала проектируемого приложения и определены связи между ними. Концептуально-инфологическая модель базы данных показана на рисунке В.6 приложения В.

2.6.2 Логическое проектирование

Между двумя или более таблицами базы данных могут существовать отношения подчиненности. Отношения подчиненности определяют, что для каждой записи исходной таблицы или так же называемой еще родительской может существовать одна или несколько записей в порождённой таблице, так же называемой еще дочерней.[8]

Существует три разновидности связей между таблицами базы данных:

- «один-ко-многим»,
- «один-к-одному»,
- «многие-ко-многим».

Отношение «один-ко-многим» имеет место, когда одной записи родительской таблицы может соответствовать несколько записей в дочерней таблице. Данная связь является самой распространенной для реляционных баз данных.

Отношение «один-к-одному» имеет место, когда одной записи в родительской таблице соответствует одна запись в дочерней таблице.

Отношение «многие-ко-многим» имеет место, когда:

- записи в родительской таблице может соответствовать больше одной записи в дочерней таблице;
- записи в дочерней таблице может соответствовать больше одной записи в родительской таблице.

Итоговые отношения с учетом добавления внешних ключей показаны на рисунках Г.1 – Г.25 приложения Г. Все отношения приведены к третьей нормальной форме, согласно техническому заданию. Все атрибуты сущностей атомарны, то есть сущности соответствуют первой нормальной форме, также сущности имеют один первичный ключ, от которого зависят все остальные атрибуты сущности, то есть все таблицы

приведены ко второй нормальной форме. Все атрибута сущностей зависят только от первичного ключа следовательно база данных приведена к третьей нормальной форме.

Отдельно стоит отметить отношение, показанное на рисунке Г.24, где для разрешения связи «многие-ко-многим» потребовалось создать дополнительную сущность совместимость. Итоговая логическая модель базы данных показана на рисунке Г.26 приложения Г.

2.6.3 Физическое проектирование

Физическое проектирование является третьим и последним этапом создания проекта базы данных, при выполнении которого проектировщик принимает решения о способах реализации разрабатываемой базы данных. Во время предыдущего этапа проектирования была определена логическая структура базы данных (которая описывает отношения и ограничения в рассматриваемой прикладной области). Хотя эта структура не зависит от конкретной целевой СУБД, она создается с учетом выбранной модели хранения данных, например реляционной, сетевой или иерархической. Однако, приступая к физическому проектированию базы данных, прежде всего необходимо выбрать конкретную целевую СУБД. В качестве системы управления базой данных была выбрана MySQL.[9]

Физическое проектирование базы данных показано в таблицах Д.1 – Д.24 приложения Д. Физическая модель базы данных показана на рисунке Д.1 приложения Д.

2.7 Выбор и обоснование средств реализации

Для реализации мобильного приложения «Продажа запчастей» был выбран современный технологический стек, обеспечивающий высокую производительность, надежность и удобство разработки. В качестве основы серверной части проекта используется фреймворк Django на языке Python. Этот выбор обусловлен его мощным функционалом для быстрой разработки сложных API-интерфейсов и отличной поддержкой работы с различными системами управления базами данных. Django REST Framework [4] дополняет этот стек, предоставляя все необходимые инструменты для

создания полноценного REST API, который будет обслуживать мобильное приложение, обеспечивая безопасный обмен данными и поддерживая все стандартные HTTPS-методы.

Выбор Django для серверной части обусловлен его принципиальными преимуществами перед альтернативными платформами. В отличие от Node.js и Express, Django предоставляет строгую Model-View-Controller архитектуру, включая мощный слой взаимодействия с базой данных, что критически важно для системы со сложной бизнес-логикой. По сравнению с PHP-фреймворками, Python и Django демонстрирует лучшую производительность при работе с большими объемами данных.

Для разработки клиентской части был выбран язык Kotlin с использованием современного UI-фреймворка Jetpack Compose, который является стандартным при разработке мобильных приложений. Kotlin обеспечивает безопасность типов, лаконичный синтаксис и полную совместимость с существующими Java-библиотеками фреймворка Android.[10] В сравнении с кроссплатформенными решениями, такими как Flutter, React Native, нативный подход к разработке обеспечивает заметно лучшую производительность и полный доступ к специфичным Android API, более плавную анимацию и лучшую интеграцию с системными сервисами. При этом Jetpack Compose, в отличие от традиционной Android-разработки на XML, значительно ускоряет создание пользовательского интерфейса благодаря современному декларативному подходу к разработке.

В качестве системы управления базами данных выбрана MySQL – надежная реляционная СУБД. Она идеально подходит для данного проекта благодаря своей стабильности, высокой производительности при работе с большими объемами данных и отличной интеграции с Django через стандартный адаптер. MySQL обеспечивает все необходимые функции для работы с каталогом товаров, пользовательскими данными и заказами, включая сложные запросы, транзакции и полноценную систему резервного копирования. Решение использовать MySQL в качестве СУБД основано на ее конкурентных преимуществах: по сравнению с PostgreSQL она предъявляет меньшие требования к ресурсам, в отличие от MongoDB лучше поддерживает сложные SQL-

запросы для аналитики, а в сравнении с SQLite предоставляет полноценные возможности масштабирования.

Выбранный технологический стек оптимально сочетает в себе производительность, надежность и скорость разработки. Python и Django на стороне сервера обеспечивают быструю реализацию бизнес-логики и API, Kotlin и Jetpack Compose позволяют создать современное высокопроизводительное мобильное приложение, а MySQL гарантирует надежное хранение и эффективный доступ ко всем данным системы. Все компоненты стека имеют активные сообщества разработчиков, регулярно обновляются и поддерживают современные стандарты безопасности, что особенно важно для приложения, работающего с персональными данными пользователей.

В ходе выполнения проектирования программного обеспечения было выполнено разбиение на отдельные модули, определены их функции и связи между ними. Выполнено инфологическое, логическое и физическое проектирование базы данных с нормализацией до третьей нормальной формы. Выбран состав технических средств для реализации проекта.

3 ОПИСАНИЕ РАЗРАБОТАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Общие сведения о программном обеспечении

Наименование программного обеспечения: «Android-приложение «Продажа запчастей»».

Автор: Студент группы 1105-Об Логинов Никита Владимирович

Дата и место разработки: 2025 год, город Благовещенск Амурской области.

3.2 Функциональное назначение

Программное обеспечение предназначено для автоматизации процессов интернет-магазина, обеспечивая комплексное решение для управления товарами, заказами, клиентскими данными и интеграцией с внешними системами. Основное назначение системы – предоставление удобного и безопасного инструмента для взаимодействия между покупателями (физическими и юридическими лицами) и администрацией магазина, а также автоматизация ключевых бизнес-процессов.

Система решает широкий спектр задач, включая управление доступом пользователей с разграничением прав, шифрование и защиту данных, что обеспечивает безопасность хранения и передачи информации. Для клиентов предусмотрены функции регистрации и авторизации, работы с личным кабинетом, где сохраняются данные о машинах для персонализированного подбора запчастей, адреса доставки и история заказов. Покупатели могут формировать заказы через корзину, выбирать способы оплаты и доставки, а также получать предварительный расчет стоимости. Для удобства пользователей реализована система уведомлений об изменении статуса заказа.

Каталог товаров предоставляет актуальную информацию о наличии, ценах и описаниях продукции, а административный раздел позволяет управлять ассортиментом, обновлять остатки и корректировать цены. Отдельный функционал предусмотрен для юридических лиц, включая специализированный личный кабинет с дополнительными возможностями.

Программа интегрируется с CRM Bitrix24, обеспечивая синхронизацию заказов и упрощение работы с клиентской базой. Однако функциональность системы имеет

ряд ограничений: она не поддерживает самостоятельное добавление товаров клиентами, не включает встроенную систему аналитики продаж и не предусматривает мультиязычной поддержки без дополнительной доработки. Кроме того, интеграция с другими CRM, кроме Bitrix24, требует индивидуальной настройки.

3.3 Логическая структура

Готовый программный продукт включает в себя: клиентскую часть, серверную и СУБД MySQL, состоящую из 25 таблиц и использующую кодировку UTF-8. Далее будет выполнено описание модулей, выполняющих основные функции.

Модуль «Catalog» отвечает за обработку пользовательских запросов к каталогу товаров. Представляет собой класс «Catalog», который наследует класс «APIView». Имеет два основных метода: «get_content» для получения содержимого каталога и «get» для обработки get-запросов и получения содержимого каталога, вызывает метод «get_content». После начала обработки модуль выполняет разбиение строки запроса на компоненты, затем выполняет их парсинг с учетом иерархии каталога, после нахождения последнего элемента он преобразует его содержимое в JSON формат и возвращает его в качестве ответа. Если по указанному пути ничего не было найдено, то возвращает ошибку 404 на клиентское устройство. Вызывается из модуля «urls».

Модуль «Product» отвечает за обработку пользовательских запросов к каталогу товаров. Представляет собой класс «Product», который наследует класс «APIView». Имеет метод «get» для обработки get-запросов и получения детальной информации о товаре. Модуль принимает код товара и выполняет по нему поиск в таблицах базы данных выполняя подгрузку всей нужной информации, после чего преобразует информацию в JSON формат и возвращает его в качестве ответа. Если по указанному коду товара ничего не было найдено, то возвращает ошибку 404 на клиентское устройство. Вызывается из модуля «urls», вызывает модуль «models».

Модуль «Search» предназначен для поиска товаров. Представляет собой класс «Search», который наследует класс «APIView». Имеет методы: «get_availability» для получения информации о наличии товара, «search» для выполнения поиска по ключе-

вому слову и «get» для обработки GET-запросов и получения результатов поиска. Модуль принимает поисковую строку и выполняет по ней поиск в названии и описании товаров, затем преобразует информацию в JSON формат и возвращает его в качестве ответа. Вызывается из модуля «urls», вызывает модуль «models».

Модуль «Order» предназначен для работы с заказами. Представляет собой класс «Order», который наследует класс «APIView». Метод «get» обрабатывает GET-запросы, проверяя наличие параметра «orderid». Если параметр указан, метод возвращает детализированную информацию о заказе. Если параметр отсутствует, метод формирует список всех заказов пользователя с краткой информацией. Для создания нового заказа используется метод «post», который принимает данные в формате JSON, в запросе ожидается содержимое корзины. Метод создает запись заказа в базе данных, связывает с ней товары из корзины, рассчитывает общую сумму и возвращает детали созданного заказа. Вспомогательный метод «get_order_list» формирует список заказов пользователя, извлекая данные из базы и преобразуя их в JSON-структуру. Метод «get_order_content» предоставляет детализацию конкретного заказа, включая информацию о каждом товаре в его составе. Вызывается из модуля «urls», вызывает модуль «models».

Модуль «crm_integration» предназначен для интеграции с CRM-системой Bitrix24. Представляет из себя два класса «Bitrix24Worker» и «ChangeOrderStatus». Первый класс отвечает за отправку уведомлений и создание заказов в CRM-системе, а второй за обработку исходящих веб-хуков. Метод «create_order» класса «Bitrix24Worker» вызывается из модуля «Orders» при создании заказа и при помощи API Bitrix24 выполняет создание записи в CRM. Метод «create_request» отвечает за отправку заявок. Вызывается из модуля «urls», вызывает модуль «models».

Модуль «Cart» предназначен для поиска товаров. Представляет собой класс «Cart», который наследует класс «APIView». Метод «get_price» служит вспомогательной функцией для определения максимальной цены товара среди доступных вариантов. Метод «add_item_to_cart» обрабатывает добавление товара в корзину. Если

товар уже присутствует, его количество увеличивается на указанное значение, в противном случае создается новая запись. Метод «get» обрабатывает GET-запросы, позволяя добавлять или удалять товары через параметры URL. Если параметры не указаны, метод возвращает текущее состояние корзины. Метод «post» принимает корзину в формате JSON и полностью перезаписывает ее содержимое, удаляя старые записи и создавая новые на основе переданных данных. Метод «get_cart» формирует JSON-представление корзины, включая информацию о товарах (идентификатор, артикул, наименование, цена, количество), их доступности на складах и общих остатках. Метод «get_quantity» корректирует количество товара в корзине, если оно превышает доступные остатки, автоматически устанавливая максимально возможное значение. Вызывается из модуля «urls», вызывает модуль «models».

Модуль «Registration» реализует функцию регистрации новых пользователей при помощи метода post. Компонент представляет собой класс «Registration» унаследованный от «APIView». При вызове метода «post», он получает список данных для создания пользователя, такие как пароль и номер телефона, далее выполняет проверку данных на существование в базе, если пользователь уже существует, то возвращает ошибку. В ином случае добавляет данные пользователя в базу, автоматически создает токен аутентификации и возвращает его на устройство пользователя, это необходимо для выполнения входа сразу после регистрации. Вызывается из модуля «urls», вызывает модуль «models».

Модуль «Payment» реализует простой функционал получения списка доступных способов оплаты. Представляет собой класс «Payment», который наследует класс «APIView». Основной функционал сосредоточен в методе «get», который обрабатывает GET-запросы. При получении запроса метод вызывает вспомогательную функцию «get_methods», извлекающую все записи из модели «PaymentMethod» и преобразующую их в JSON-формат. Каждый способ оплаты представляется в виде объекта с двумя полями: «id» (уникальный идентификатор) и «methodname» (название способа оплаты). Полученные данные упаковываются в структуру с ключом «content», содержащим массив всех доступных методов.

Компонент «notification» предназначен для отправки уведомлений на клиентские устройства, реализует функционал методом «send_push_notification», принимающий в качестве аргументов код устройства, заголовок сообщения и его текст. Для отправки уведомлений используется класс «FCMNotification» библиотеки «pushcm».

Модуль «Authorize» предоставляет функционал для аутентификации пользователей в системе. Этот модуль выполняет проверку учетных данных пользователя и выдает токен доступа при успешной авторизации. При получении запроса модуль извлекает из тела запроса JSON-данные, содержащие «phone» (используемый как имя пользователя) и пароль. Если обязательные поля отсутствуют или формат запроса некорректен, система немедленно возвращает ответ со статусом 400 «Плохой запрос». Далее происходит проверка учетных данных через стандартный механизм аутентификации. В случае неудачной проверки возвращается ошибка с поясняющим сообщением о неверном имени пользователя или пароле. При успешной аутентификации модуль генерирует новый токен доступа, предварительно удаляя существующий (если он есть), чтобы обеспечить безопасность и предотвратить использование устаревших ключей. Вызывается из модуля «urls», вызывает модуль «models».

Модуль «Garage» реализует функционал управления автомобилями пользователя в системе. Модуль поддерживает ключевые операции: удаление автомобиля по идентификатору, просмотр всего списка машин и добавление новых транспортных средств. Для добавления новых машин компонент также реализует функции получения моделей машин и их производителей по запросу.

При обработке GET-запросов система проверяет наличие параметра «id» и «type» – в зависимости от их значений компонент возвращает либо список производителей, либо список моделей производителя. Если ни один из параметров явно не задан, то возвращается полный перечень машин пользователя в формате JSON, где каждая запись содержит идентификатор, ВИН-код, название производителя и модель автомобиля.

POST-запросы предназначены для добавления новых автомобилей и принимают JSON-объект с обязательными полями ВИН-кода и идентификатора модели.

Автомобиль автоматически привязывается к текущему пользователю через поле «userid».

Модуль «Address» отвечает за управление адресами доставки в системе. Представляет собой класс «Address», который наследует класс «APIView». Основной функционал модуля включает операции просмотра, добавления, редактирования и мягкого удаления (деактивации) адресных записей, связанных с текущим пользователем.

При обработке GET-запросов модуль анализирует параметры «action» и «addressid» – если указано действие удаления, система помечает соответствующий адрес как неактивный через установку флага «isactive». В остальных случаях возвращается полный список активных адресов пользователя в формате JSON, где каждый адрес содержит идентификатор, регион, город, а также подробную информацию о транспортной компании (идентификатор, название и описание).

POST-запросы обрабатывают создание и обновление адресов. Входящий JSON должен содержать обязательные поля региона, города и идентификатора транспортной компании, которые дополняются автоматической привязкой к текущему пользователю. Успешное выполнение операций добавления/изменения подтверждается пустым JSON-ответом со статусом 200.

Компонент «TransportCompany» отвечает за предоставление данных о доступных транспортных компаниях. При вызове метода «get» запрашивает данные из базы и преобразует их в JSON-формат. Предоставляет такие данные как наименование компании и ее уникальный код.

Модуль маршрутизации «urls» связывает URL-адреса с соответствующими обработчиками на основе классов Django REST Framework. Основной функционал охватывает ключевые аспекты работы интернет-магазина автозапчастей. После получения URL-запроса модуль ищет соответствующие модули обработчики и передает управление, если такой модуль не найден, то возвращает ошибку 404 «Не найдено». Вызывается из модуля «urls», вызывает модуль «models». Реализован на языке программирования Python и фреймворке Django Rest Framework.

Модуль «ApiPermission» представляет собой middleware-прослойку для Django, реализующую механизм проверки прав доступа к API. Его основная задача – выполнять фильтрацию входящих запросов обеспечивая безопасность системы. При получении запроса активируется процедура проверки доступа через вызов модуля «permission.check_access», куда передаются заголовки запроса. В случае отсутствия необходимых прав middleware немедленно прерывает обработку запроса, возвращая ответ с кодом 403 «Доступ запрещен». Только успешно прошедшие проверку запросы передаются в обработку модулю «urls». Вызывается из модуля «urls», вызывает модуль «models». Реализован на языке программирования Python и фреймворке Django Rest Framework.

Все вышеперечисленные модули относятся к серверной части и реализованы на языке программирования Python и фреймворке Django Rest Framework. UML-диаграмма классов серверной части показана на рисунке E.1 приложения E.

Компонент «OrderWorker» представляет класс «OrderWorker» унаследованный от класса «ApiConnector», UML-диаграмма которого показана на рисунке 5. UML-диаграмма классов модуля показана на рисунке 6.

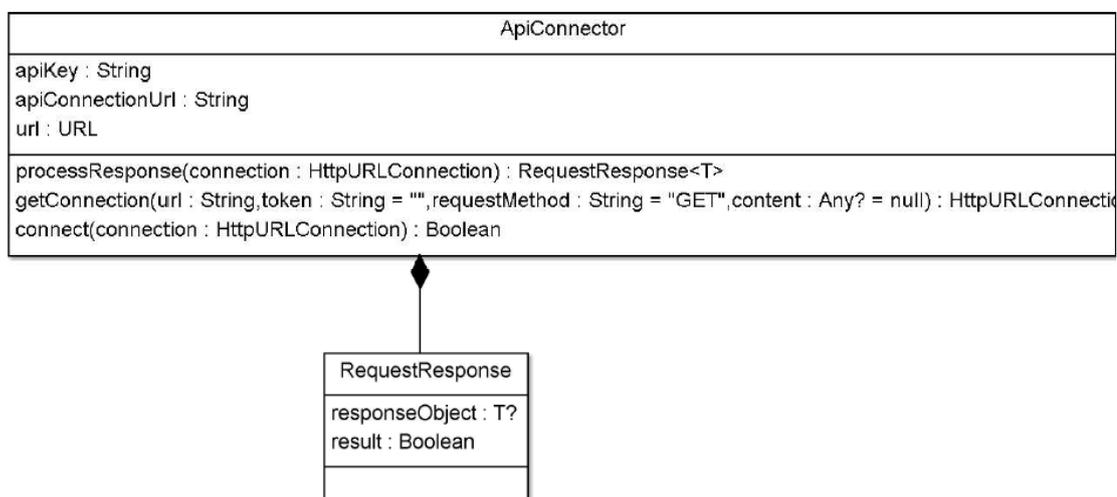


Рисунок 5 – UML-диаграмма класса «ApiConnector»

Задачи модуля заключаются в формировании запросов, связанных с заказами пользователя. Функционал реализован методами: «getOrders» для получения всех заказов пользователя, «getDetail» для детальной информации по заказу, «createOrder»

для создания заказа и «cancelOrder» для отмены заказа. При получении запроса модуль выполняет запрос к серверу, далее при помощи методов базового класса выполняет преобразование ответа из JSON-данных, полученных от сервера. Компонент вызывается из модуля «ProfileOrdersViewModel» или «CartViewModel».

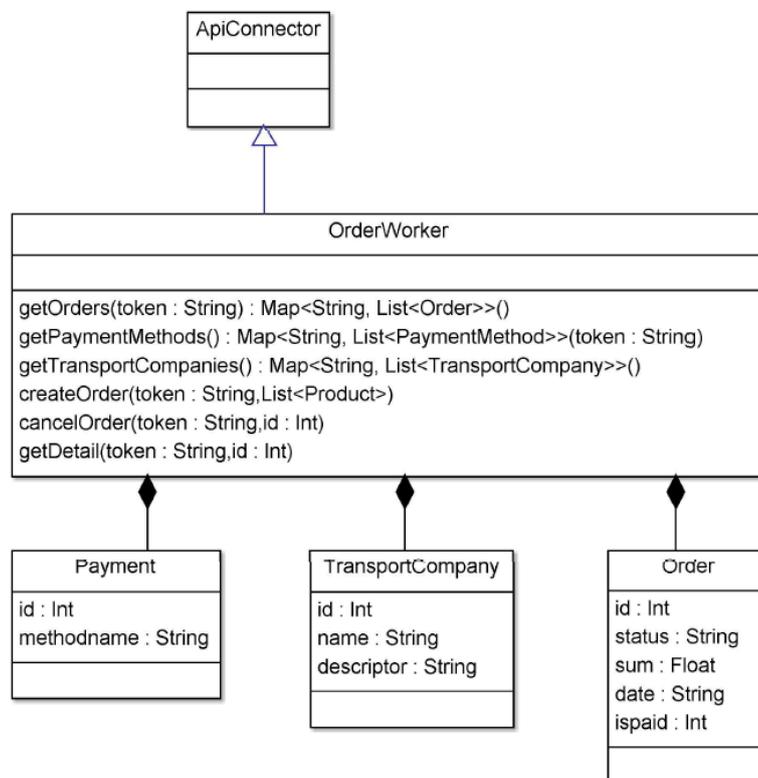


Рисунок 6 – UML-диаграмма классов модуля «OrderWorker»

Компонент «CatalogWorker» представляет собой класс, унаследованный от базового класса «ApiConnector». UML-диаграмма классов модуля показана на рисунке 7. Выполняет функции по поиску товаров, предоставлению детальной информации о товаре и содержимого категорий. Функции модуля реализованы методами: «getCatalog» для получения содержимого каталога товаров по пути, «getDetail» для получения детальной информации по коду товара, «search» для поиска, по ключевым словам, и «addToCart» для добавления запчастей в корзину.

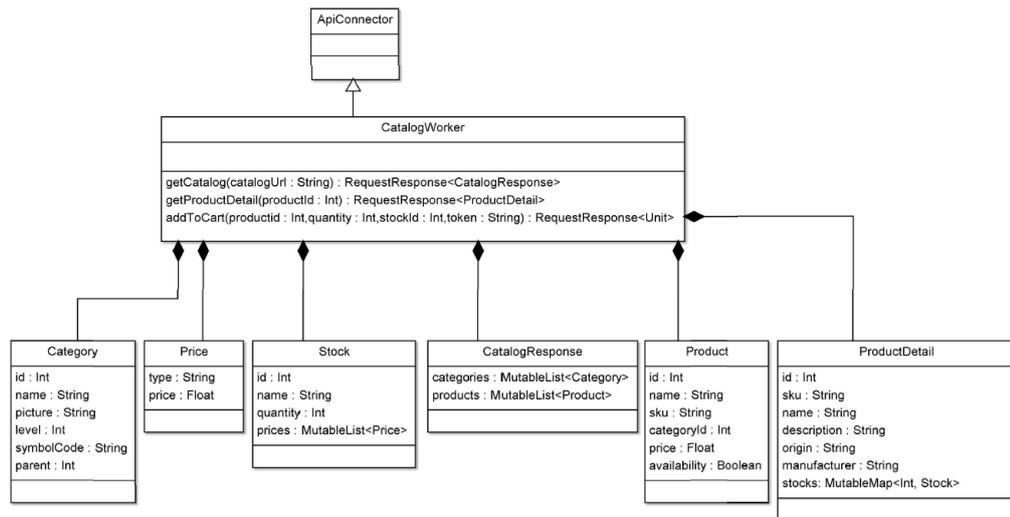


Рисунок 7 – UML-диаграмма классов модуля «CatalogWorker»

Модуль «CartWorker» представляет собой класс с аналогичным названием, унаследованный от класса «ApiConnector», выполняющий запросы для работы с корзиной запчастей.

Основной функционал реализован методами: «getCart» для получения содержимого корзины, «saveCart» для сохранения состояния корзины, «addProduct» для добавления товара в корзину. После каждого запроса методы возвращают актуальное состояние корзины и код ответа. Для хранения данных используются два дата-класса: «Product» и «Cart».

Класс «AddressWorker» представляет собой клиентский модуль для работы с адресами доставки, реализованный как наследник базового класса «ApiConnector». Основу модуля составляет дата-класс «Address», который хранит всю необходимую информацию об адресе: уникальный идентификатор (опциональный для новых адресов), регион и город доставки, а также детали транспортной компании (идентификатор, название и описание). Этот класс служит универсальной моделью для передачи данных между клиентом и API.

Главный метод «getAddresses» выполняет запрос на получение всех адресов пользователя, возвращая их в виде словаря, где ключами выступают идентификаторы адресов, а значениями соответствующие объекты «Address». Для удаления адресов

используется метод «deleteAddress», который принимает идентификатор адреса и формирует соответствующий запрос к API. Наиболее универсальный метод «createOrUpdateAddress» позволяет как создавать новые адреса (когда addressid равен null), так и обновлять существующие, передавая полный объект Address в теле POST-запроса. Модуль вызывается из модуля «ProfileAddressViewModel» или «OrderViewModel».

Класс «GarageWorker» представляет собой модуль для управления гаражом пользователя. Наследуется от базового класса «ApiConnector». UML-диаграмма классов модуля показана на рисунке 8.

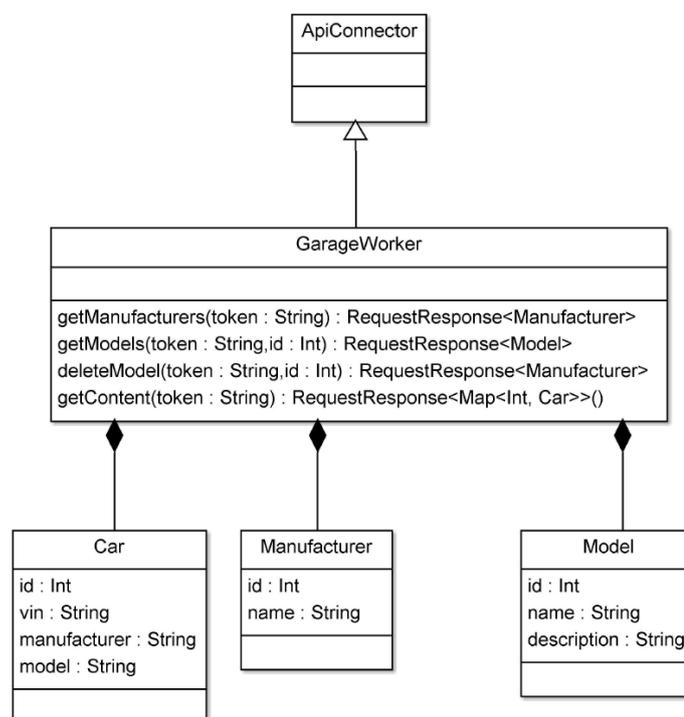


Рисунок 8 – UML-диаграмма классов модуля «GarageWorker»

Основу модуля составляют три дата-класса: «Car» для хранения информации об автомобилях: идентификатор, ВИН-код, производитель и модель, «Manufacturer» для работы с производителями и «Model» для представления конкретных моделей автомобилей. Эти классы точно отражают структуру данных API и обеспечивают строгую типизацию при работе с ответами сервера.

Функциональность обеспечивается четырьмя методами: получение списка производителей через «getManufacturers», загрузку моделей конкретного производителя с помощью «getModels», удаление модели по идентификатору через «deleteModel» и получение всего содержимого гаража пользователя с использованием «getContent». Компонент вызывается из модуля «ProfileGarageViewModel».

Все вышеперечисленные модули относятся к клиентской части приложения и реализованы на языке программирования Kotlin и фреймворке Jetpack Compose.

3.4 Условия применения

Для функционирования программного обеспечения требуется сервер, хостинг с установленным веб-сервером или VDS сервер. Серверной частью должна использоваться операционная система Ubuntu версии 23.04 или выше.

Минимальные требования к техническим средствам сервера:

- процессор с тактовой частотой не ниже 2 ГГц;
- оперативная память от 2 ГБ;
- свободное дисковое пространство от 10 ГБ;
- установленная и настроенная СУБД (MySQL);
- панель управления хостингом ipsmanager;
- интерпретатор Python версии 3.11 или выше.

Для функционирования клиентской части требуется устройство с операционной системой Android версии API 30 (Android 11) или выше. Для функционирования ПО устройство должно быть подключено к сети «Интернет». Минимальные требования к техническим средствам клиентского устройства:

- процессор с тактовой частотой не ниже 1 ГГц;
- оперативная память от 500 Мб;
- свободное дисковое пространство не менее 100 Мб.

Требования к прикладному ПО для клиентской части отсутствуют так как все необходимые библиотеки будут являться частью разрабатываемого ПО и не потребуют дополнительной установки.

3.5 Настройка программы

Клиентская часть программы поставляется в виде файла с наименованием «tps_trade.apk», где «.apk» – это расширение установочного файла для ОС Android. Для установки необходимо открыть файл на устройстве, нажать кнопку «установить» в появившемся диалоговом окне. После завершения установки можно начинать работу с приложением.

Серверная часть приложения поставляется в виде двух файлов:

- архив под названием «tps_trade.zip» содержащий папку с файлами;
- дамп базы данных MySQL под названием «tps_trade.sql» с основным содержанием.

В первую очередь необходимо создать БД на сервере, для этого в панели управления iprmanager нужно перейти во вкладку «Базы данных», открыть вкладку «Дамп» и выбрать файл «tps_trade.sql» из файлов, загруженных на сервер или на компьютере. После этого в списке баз данных появится база «tps_trade», для которой нужно выполнить создание нового пользователя с правами на создание, удаление, изменение записей в БД с доступом ко всем таблицам, имя пользователя значения не имеет, пароль должен быть достаточно сложным чтобы исключить несанкционированный доступ.

Перед установкой программы необходимо подключить интерпретатор python, для этого во вкладке «Модули» выполняется его установка.

Далее во вкладке «Сайты» выполняем создание нового сайта, путем нажатия на кнопку «Создать сайт», со следующими настройками:

- доменное имя: ip-адрес сервера;
- директория сайта: www/tps_trade;
- обработчик: python;
- режим работы приложения: web-сервер python;
- версия python: 3.11;
- база данных: tps_trade;
- аргументы сервера: python tps_trade/manage.py runserver.

После создания сайта нужно нажать кнопку «Конфиг. файлы», в открывшемся текстовом редакторе ввести команду «django» и сохранить файл. Далее через Shell-клиент сервера вводятся следующие команды, для установки всех необходимых библиотек и активации виртуального пространства:

- source .venv/bin/activate
- pip install --upgrade pip
- pip install wheel
- pip install mysqlclient

Затем содержимое архива помещается в папку «www/tps_trade», после чего необходимо перейти к непосредственной настройке программы. В файле «settings.py» выполняются следующие настройки:

- ALLOWED_HOSTS = [ip-адрес сервера];
- в пункте DATABASES – ENGINE: «django.db.backends.mysql», NAME: «tps_trade», HOST: «localhost», USER: имя созданного пользователя, PASSWORD: пароль созданного пользователя.

После всех выполненных действий выполняется перезапуск сервера для сохранения настроек.

3.6 Обращение к программе

Запуск серверной части программы происходит при помощи командной строки ОС Ubuntu введением команды вида «python «путь к папке сервера/manage.py» runserver». Ключевое слово «python» указывает операционной системе что файл необходимо запустить с использованием установленного интерпретатора языка Python. Команда «runserver» дает команду на запуск сервера приложения.

Запуск клиентской части программы происходит путем нажатия на ярлык на главном экране клиентского устройства. Дополнительных действий при этом не требуется.

3.7 Основные экранные формы приложения

Для оформления заказа в мобильном приложении, для начала необходимо собрать нужные товары в корзину. Для этого нужно перейти во вкладку «Каталог» и

выбрать при помощи фильтров или поиска нужные запчасти. Добавить запчасть в корзину, если она в наличии, можно при помощи страницы детального просмотра, где есть возможность не только добавить товар, но и выбрать склад отгрузки. Каталог товаров и детальная страница показаны на рисунке 9.

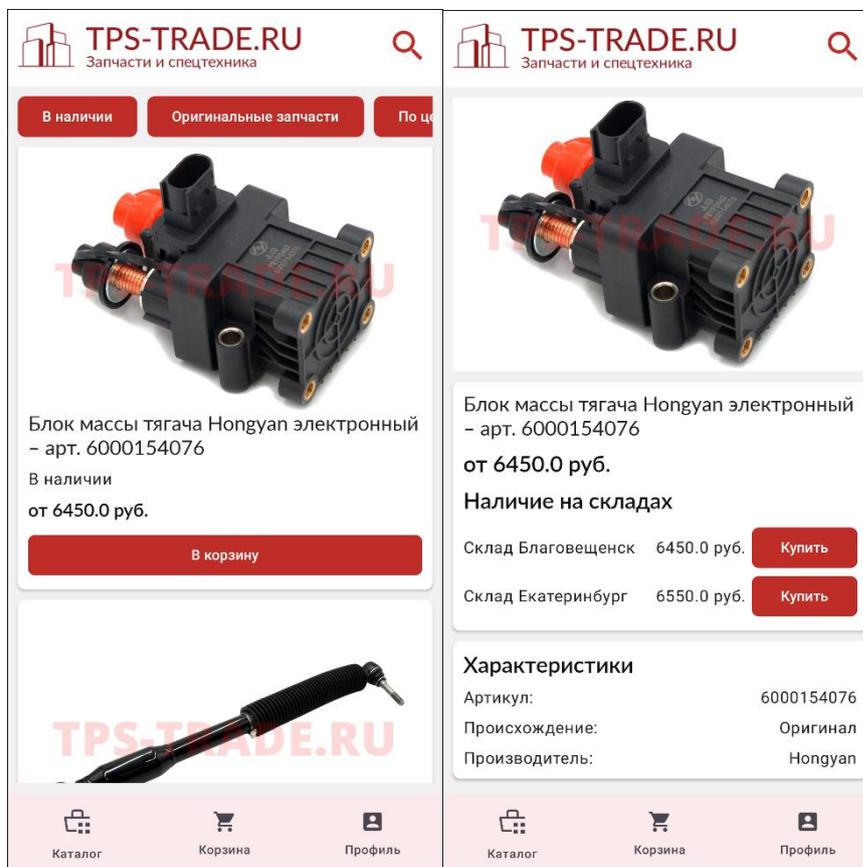


Рисунок 9 – Каталог и детальная страница товара

После выбора запчастей необходимо перейти во вкладку «Корзина» для оформления заказа, тут же можно удалить ошибочно добавленный товар или изменить их количество. Для оформления заказа нужно нажать кнопку «Оформить заказ» или «Быстрый заказ», в этом случае заказ мгновенно оформится и пользователю не придется вводить лишние данные, они будут уточнены менеджером позже. В ином случае пользователю отобразится последовательность форм для выбора адреса, транспортной компании и способа оплаты. Формы показаны на рисунках 10 и 11.

Шаг 1: Выбор адреса	Шаг 2: Выбор способа доставки
Выберите адрес доставки	Выберите способ доставки
Регион: Амурская область	СДЭК <input type="radio"/>
Город: Благовещенск	Деловые линии <input type="radio"/>
Адрес: ул. Ленина, 197	ПЭК <input checked="" type="radio"/>
Добавить адрес	Самовывоз <input type="radio"/>
	ТК Энергия <input type="radio"/>
	Другая ТК <input type="radio"/>
	Байкал Сервис <input type="radio"/>
	Флагман <input type="radio"/>
Далее	Далее
Каталог Корзина Профиль	Каталог Корзина Профиль

Рисунок 10 – Выбор транспортной компании и адреса

После оформления заказа он отобразится в списке в личном кабинете пользователя, во вкладке «Заказы». Тут же можно проверить его содержимое, отследить статус или отменить заказ. Вкладка «Заказы» показана на рисунке 12.

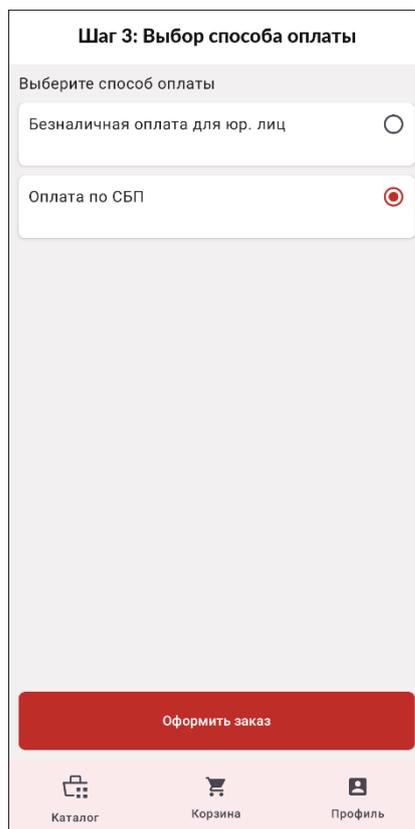


Рисунок 11 – Выбор способа оплаты

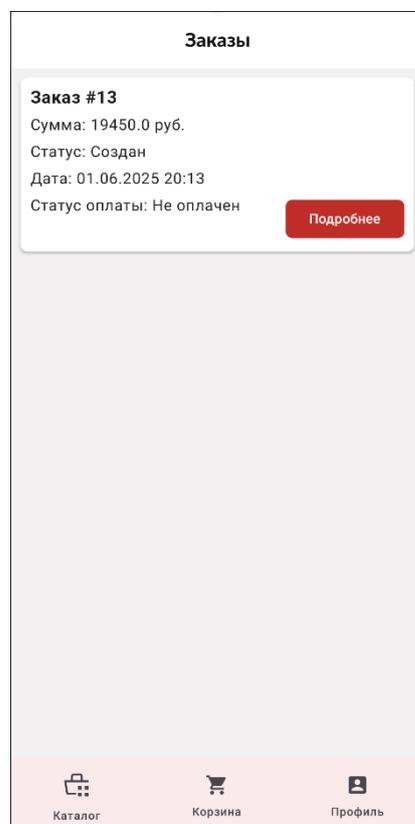


Рисунок 12 – Вкладка «Заказы» личного кабинета пользователя

4 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

Разработка мобильного приложения «Продажа запчастей» для ООО «УникОпт» требует комплексного подхода, охватывающего не только функциональные и технические характеристики, но и три ключевых аспекта: безопасность пользователей, экологическую ответственность и устойчивость к чрезвычайным ситуациям. В данном разделе детально рассматриваются все эти вопросы и их соответствие нормативным требованиям.

4.1 Безопасность

Разработанное приложение предназначено для клиентов ООО «УникОпт», которые будут активно использовать его в течение рабочего дня. Важно минимизировать негативное воздействие на здоровье пользователей, связанное с длительным использованием мобильных устройств.

Работа с мобильными устройствами создает специфическую нагрузку на организм. Напряженный характер труда проявляется в необходимости постоянной концентрации внимания на небольшом экране, выполнении монотонных операций и поддержании статичной позы при работе с устройством. Эти факторы могут приводить к зрительному утомлению, мышечному напряжению в шейном отделе и кистях рук, а также психоэмоциональному истощению.

К основным рискам относятся производственные факторы, связанные с вредными условиями труда, включая чрезмерные физические и психоэмоциональные нагрузки, а также работу в условиях воздействия опасных производственных факторов. Особое внимание следует уделять экологическим угрозам, таким как загрязнение окружающей среды токсичными веществами, наличие вредных химических соединений в воздухе, воде и пищевых продуктах, а также накопление опасных веществ в производственной среде. Не менее важны физические воздействия, включая акустическое загрязнение, тепловое и электромагнитное излучение от технических устройств, а также возможное воздействие ионизирующих излучений. Для снижения

влияния все вышеперечисленных негативных факторов рабочее место должно соответствовать действующим санитарным нормам и требованиям охраны труда.

4.1.1 Требования к мобильным устройствам

Безопасность пользователей мобильного приложения «Продажа запчастей» во многом зависит от характеристик устройств, на которых оно будет работать. Поскольку сотрудники и клиенты ООО «УникОпт» будут активно использовать приложение в течение рабочего дня, необходимо учитывать как санитарно-гигиенические нормы, так и технические требования к мобильным устройствам.

Важным аспектом является контроль электромагнитного излучения. Согласно СанПиН 2.1.3685-21, плотность потока энергии в частотном диапазоне 0,3–300 ГГц не должна превышать 10 мкВт/см². Для снижения воздействия излучения рекомендуется использовать гарнитуры или громкую связь вместо длительного удержания устройства у головы, а также избегать ношения телефона в карманах одежды.

Экран мобильного устройства должен обеспечивать комфортную работу без излишней нагрузки на зрение. Яркость дисплея должна автоматически регулироваться в зависимости от освещения или настраиваться вручную в диапазоне 150–300 кд/м². Частота обновления экрана не должна быть ниже 60 Гц для плавного отображения контента. Для защиты глаз от синего света, особенно при работе в вечернее время, желательно использовать встроенные фильтры. Размер шрифта в приложении не должен быть меньше 14 пунктов, чтобы исключить необходимость приближать устройство к глазам.

Эргономика также играет ключевую роль в обеспечении комфортной работы. Для предотвращения напряжения в шее и спине рекомендуется использовать подставки для устройств, а при длительном вводе данных – внешние клавиатуры. Сотрудникам следует делать регулярные перерывы каждые 45-60 минут, чтобы снизить нагрузку на зрение и опорно-двигательный аппарат.

Пользователям рекомендуется избегать работы с устройством при уровне заряда батареи ниже 20%, так как это может привести к снижению яркости экрана и увеличению нагрузки на глаза. Регулярная очистка экрана от загрязнений улучшает

читаемость, а настройка напоминаний о перерывах поможет соблюдать режим работы. Соблюдение этих требований обеспечит безопасность и комфорт при использовании разработанной программы, а также минимизирует негативное воздействие на здоровье пользователей.

4.1.2 Требования к организации рабочего пространства

Организация рабочего пространства для пользователей, использующих приложение «Продажа запчастей», требует комплексного подхода, учитывающего как особенности работы с мобильными устройствами, так и общие требования к офисным рабочим местам. При длительной работе с приложением важно создать условия, минимизирующие физическую и зрительную нагрузку, что напрямую влияет на производительность труда и здоровье персонала.

Рабочее место должно располагаться так, чтобы естественный свет падал сбоку, предотвращая блики на экране. При искусственном освещении необходимо обеспечить равномерный свет 300-500 люкс с температурой 4000-5000 К. Избегайте резких контрастов между яркостью экрана и окружающего пространства, чтобы снизить нагрузку на зрение.

Рабочее место необходимо организовать для снижения нагрузки на шею и спину. При длительной работе с мобильным устройством используйте подставку, располагая экран на уровне глаз. Для ввода текста применяйте внешнюю клавиатуру, установленную на стандартной высоте 72-75 см. Кресло должно поддерживать поясницу и регулироваться так, чтобы ноги стояли полностью на полу, а бёдра были параллельны полу.

Микроклиматические условия в помещении играют важную роль в обеспечении комфортной работы. Температура воздуха должна поддерживаться в диапазоне 22-24 °С в холодный период года и 23-25 °С в тёплый период при относительной влажности 40-60 %. Скорость движения воздуха не должна превышать 0,1 м/с. Особое внимание следует уделить качеству воздуха – регулярное проветривание или использование систем кондиционирования помогает снизить концентрацию углекислого газа.

Акустический комфорт рабочего места предполагает уровень фонового шума не более 55 дБА. Для снижения шумовой нагрузки рекомендуется использовать звукопоглощающие материалы в отделке помещения. При этом следует учитывать, что длительное использование наушников может вызывать дополнительную усталость, поэтому необходимо регламентировать время их непрерывного применения.

Эргономика рабочего пространства должна учитывать необходимость периодической смены позы и вида деятельности. Рекомендуется оборудовать зоны для кратковременного отдыха, где сотрудники могут сменить положение тела, выполнить простые физические упражнения или гимнастику для глаз. Особое значение имеет организация перерывов – после каждого часа работы с мобильным устройством следует делать 5-10 минутный перерыв.

Безопасность рабочего места требует свободного пространства – не менее 6 м² и аккуратного размещения проводов, чтобы избежать травм и повреждений. Важно соблюдать пожарную безопасность: не оставлять заряжающиеся устройства без присмотра и использовать только сертифицированное оборудование. Регулярные проверки условий труда помогают вовремя устранять проблемы.

Правильная организация рабочего требует комплексного подхода, учитывающего как технические аспекты работы с устройствами, так и физиологические потребности человека. Соблюдение этих требований позволит не только повысить эффективность работы сотрудников, но и сохранить их здоровье при длительном использовании мобильных технологий.

4.1.3 Требования к освещению на рабочих местах

Организация правильного освещения рабочего места для сотрудников представляет собой сложную задачу, требующую учета множества факторов. Оптимальное освещение должно обеспечивать не только комфортную работу с цифровыми устройствами, но и предотвращать зрительное утомление, которое может возникать при длительном использовании мобильных гаджетов. Особенность современных ра-

бочих мест заключается в том, что они часто сочетают работу с бумажными документами, компьютером и мобильными устройствами, что предъявляет особые требования к системе освещения.

Особое внимание следует уделить характеристикам светового потока. Наиболее комфортным для глаз считается свет с цветовой температурой 4000-5000 К, который обеспечивает нейтральное освещение. Такой свет не искажает цветопередачу на экране мобильного устройства и одновременно создает комфортную атмосферу для работы. Коэффициент пульсации освещения должен быть минимальным – не более 5%, так как даже незначительные колебания светового потока, незаметные глазу, могут вызывать повышенную утомляемость при длительной работе.

Регулярное обслуживание осветительных систем является обязательным условием поддержания оптимальных параметров освещения. В помещениях ООО «УникОпт» используется два вида освещения: естественное и искусственное. Окна в помещениях офиса ориентированы на все четыре стороны света, что обеспечивает высокий уровень естественного освещения. Все требования к освещению выполняются.

4.1.4 Требования к микроклимату на рабочих местах

В отличие от стационарных рабочих мест с ПК, работа с мобильными устройствами часто происходит в различных условиях – от офисных помещений до складских комплексов и торговых залов, что предъявляет особые требования к созданию комфортной и безопасной среды.

В офисных помещениях, где сотрудники преимущественно работают с приложением, должны поддерживаться оптимальные параметры микроклимата в соответствии с требованиями СанПиН 2.1.3685-21. Для категорий работ Ia и Ib, к которым относится деятельность по вводу данных и работе с мобильными устройствами, установлены следующие параметры: в холодный период года температура воздуха должна составлять 22-24 °С для категории Ia и 21-23 °С для категории Ib; в теплый период – 23-25 °С и 22-24 °С соответственно.

Особое внимание уделяется содержанию аэроионов в воздухе рабочих помещений. Современные устройства могут влиять на ионный состав воздуха. Концентрация

легких аэроионов должна поддерживаться в пределах 400-50000 ионов/см³ для отрицательных ионов и 400-50000 ионов/см³ для положительных, как того требуют санитарные нормы.

Содержание вредных химических веществ в воздухе рабочих помещений не должно превышать предельно допустимых концентраций, установленных СанПиН 2.1.3685-21. Особый контроль осуществляется за такими веществами, как формальдегид (ПДК 0,05 мг/м³), фенол (0,01 мг/м³), стирол (0,04 мг/м³), которые могут выделяться из материалов корпусов электронных устройств и офисной мебели при повышенных температурах.

Система вентиляции в офисных помещениях спроектирована с учетом повышенных требований к воздухообмену в зонах интенсивного использования электронных устройств. Кратность воздухообмена составляет не менее 60 м³/ч, что позволяет эффективно удалять возможные вредные выделения и поддерживать оптимальный кислородный баланс.

В офисе, арендуемом ООО «УникОпт», все параметры микроклимата и качества воздуха соответствуют всем требованиям.

4.1.5 Требования к организации и оборудованию рабочих мест с ПК

Основные рабочие места в офисе оборудованы с учетом необходимости комбинированной работы – как с мобильными устройствами, так и со стационарной техникой. Столы имеют глубину 700-800 мм и ширину 1200-1400 мм, что обеспечивает достаточно места для размещения документов и дополнительного оборудования. Особое внимание уделено организации пространства для ног – обеспечен свободный участок глубиной не менее 450 мм и высотой 600 мм под столом.

Для работы с мобильными устройствами используются эргономичные подставки с регулируемым углом наклона от 15 до 70 градусов, позволяющие установить гаджет на оптимальной высоте – примерно на уровне глаз пользователя, то есть на высоте 90-110 см от пола. Расстояние от глаз до экрана устройства составляет 40-50 см для смартфонов и 50-70 см для планшетов.

Офисные кресла соответствуют современным эргономическим стандартам. Они обеспечивают:

- регулировку высоты сиденья в диапазоне 42-52 см;
- изменение угла наклона спинки от вертикального положения до 30 градусов назад;
- возможность фиксации глубины сиденья в пределах 38-44 см;
- регулируемые подлокотники по высоте (18-25 см от сиденья) и ширине.

Особое внимание уделено безопасности при одновременном использовании нескольких устройств. Минимальное расстояние между рабочими местами составляет 1,2 м по фронту и 2 м в глубину, что исключает взаимные помехи и обеспечивает комфортные условия работы.

В ходе проверки рабочих мест в ООО «УникОпт» было установлено, что все требования к организации пространства для работы выполняются.

4.1.6 Нормирование работы за ПК и перерывов

При разработке мобильного приложения особое внимание уделяется вопросам нормирования рабочего времени сотрудников, использующих мобильные устройства в своей профессиональной деятельности. В отличие от стационарных рабочих мест с ПК, работа с мобильными устройствами имеет свои особенности, которые необходимо учитывать при организации трудового процесса и планировании перерывов.

Основой для нормирования рабочего времени служит оценка интенсивности и характера труда сотрудников, работающих с мобильным приложением. Все виды деятельности можно разделить на три основные категории. Категория «А» включает операции по просмотру и анализу информации на экране устройства. Категория «Б» охватывает активный ввод данных. Категория «В» объединяет творческие задачи, выполняемые в интерактивном режиме.

Особенностью работы с мобильными устройствами является более высокая нагрузка на зрительный аппарат из-за меньшего размера экрана и необходимости постоянной фокусировки зрения. Поэтому для каждой категории работ установлены следующие ограничения: для группы «А» – не более 60000 знаков за смену, для

группы «Б» – до 40000 знаков, для группы «В» – не более 6 часов непрерывной работы. Эти нормативы учитывают повышенную утомляемость при работе с мобильными устройствами и направлены на сохранение здоровья сотрудников.

Временные нормы регламентированных перерывов для работы с мобильным приложением увеличены на 15-20 % по сравнению с аналогичными показателями для ПК. Для 8-часовой рабочей смены суммарное время перерывов составляет: 60 минут для I категории, 85 минут для II категории и 110 минут для III категории. При 12-часовой смене эти показатели увеличиваются до 95, 130 и 165 минут соответственно. Такой подход учитывает дополнительные факторы утомления, связанные с использованием мобильных устройств – статическую нагрузку на шейный отдел позвоночника и необходимость частой перефокусировки зрения.

В ООО «УникОпт» для контроля соблюдения норм рабочего времени и перерывов может использоваться существующая CRM-система, но фактически контроль не осуществляется.

4.1.7 Требования к графическому интерфейсу приложения

Графический интерфейс мобильного приложения «Продажа запчастей» разрабатывается с учетом современных стандартов эргономики и требований безопасности труда, регламентированных ГОСТ Р 52872-2019. Особое внимание уделялось созданию интуитивно понятного и визуально комфортного интерфейса.

Структура интерфейса строится по принципу функциональных зон, четко разделяющих области навигации, каталога товаров, корзины заказов и служебных функций. Каждый раздел имеет визуальные отличия в оформлении, но сохраняет общую стилистику интерфейса, что обеспечивает логичность взаимодействия.

Типографика интерфейса соответствует строгим требованиям читаемости. Основной шрифт обеспечивает высоту символов не менее 3,2 мм. Важные элементы интерфейса выделены полужирным начертанием. Приложение поддерживает функцию динамического изменения размера шрифта адаптации под индивидуальные потребности пользователей. Все элементы интерфейса масштабируются пропорционально, сохраняя логические связи между компонентами.

Таким образом, разработанное мобильное приложение соответствует всем требованиям к графическому интерфейсу.

4.2 Экологичность

Разработка и внедрение мобильного приложения осуществляется с учетом принципов экологической ответственности и устойчивого развития. Несмотря на виртуальную природу программного продукта, его жизненный цикл оказывает прямое и косвенное воздействие на окружающую среду через использование вычислительных ресурсов, эксплуатацию мобильных устройств и изменение бизнес-процессов компании. В соответствии с требованиями федерального закона №89 «Об отходах производства и потребления», особое внимание уделяется вопросам минимизации экологического следа на всех этапах – от проектирования архитектуры до утилизации оборудования.

Экологическая стратегия проекта строится на трех ключевых направлениях: энергоэффективность, сокращение материальных ресурсов и ответственное обращение с отходами. Техническая реализация приложения использует современные подходы к оптимизации кода, что позволяет снизить энергопотребление как на стороне серверного оборудования, так и на мобильных устройствах сотрудников. Алгоритмы работы минимизируют нагрузку на процессор и объем передаваемых данных, что особенно важно для мобильных сетей с их повышенным энергопотреблением. Серверная часть размещается в облачной инфраструктуре с динамическим масштабированием ресурсов, что обеспечивает оптимальное использование вычислительных мощностей без избыточного резервирования. Энергосберегающие технологии реализованы и на уровне пользовательского интерфейса разработанного приложения. Адаптивная регулировка яркости и ограничение ресурсоемких анимаций позволяют продлить срок автономной работы мобильных устройств.

Особое значение имеет вклад ЭДО в цифровизацию документооборота компании. Полный переход на электронные накладные, счета-фактуры и акты выполненных работ позволит сократить потребление бумаги на 80%. Встроенные инструменты

электронной подписи и шифрования данных обеспечивают юридическую значимость документов без использования бумажных носителей.

Особое внимание уделяется замене устаревшего оборудования на энергоэффективные аналоги. В офисных помещениях проведен полный переход на светодиодное освещение, что позволило исключить использование ртутьсодержащих люминесцентных ламп I класса опасности. Новые светильники потребляют меньше электроэнергии и имеют больший срок службы.

К сожалению, в организации не осуществляется правильная утилизация оборудования, после завершения срока эксплуатации или поломки оно уходит на хранение или утилизируется вместе с общими отходами. Такой подход не является правильным, оптимальным решением в этой ситуации будет поиск специальной организации, осуществляющей вывоз и правильную утилизацию оборудования.

4.3 Чрезвычайные ситуации

Основой системы безопасности компании являются требования Федерального закона №123 «Технический регламент о требованиях пожарной безопасности» и Постановления Правительства №1479 от 16.09.2020. В офисных помещениях ООО «УникОпт» обеспечены все необходимые меры пожарной безопасности. Помещения оборудованы современной автоматической системой пожарной сигнализации с датчиками дыма и тепловыми извещателями, подключенной к центральному пульту наблюдения. Система оповещения включает как звуковые, так и световые сигналы. Все эвакуационные пути четко обозначены светящимися указателями и поддерживаются в свободном состоянии, что особенно важно при эвакуации в условиях задымления.

Важной составляющей безопасности является регулярное обучение персонала. Все сотрудники ООО «УникОпт», проходят обязательный противопожарный инструктаж и инструктаж по технике безопасности с периодичностью не реже одного раза в год.

ЗАКЛЮЧЕНИЕ

При выполнении ВКР были выполнены следующие этапы:

- проведен анализ рынка и потребностей клиентов ООО «УникОпт»;
- разработаны функциональные требования к приложению и техническое задание;
- была выбрана архитектура и спроектирован пользовательский интерфейс;
- реализован функционал приложения;
- было выполнено тестирование программного средства.

На этапе анализа была детально изучена организационная структура и бизнес-процессы компании, были выявлены узкие места в текущей системе продаж. В ходе анализа существующей ИКТ-инфраструктуры предприятия и были выявлены ее сильные стороны, которые можно использовать для расширения функционала приложения и дополнительной оптимизации бизнес процессов.

Результатом проектирования стал выбор клиент-серверной архитектуры приложения, включающей оптимизированную структуру базы данных и детализированные схемы взаимодействия компонентов. Особое внимание было уделено созданию гибкой архитектуры, способной адаптироваться к изменяющимся бизнес-потребностям компании.

На этапе разработки было создано полностью работоспособное программное обеспечение, соответствующее техническому заданию. Готовое приложение включает в себя 15 модулей серверной части, 15 модулей клиентской, 22 таблицы базы данных и более 20 экранных форм. Программа была реализована при помощи Django Framework и Jetpack Compose.

В ходе тестирования было выявлено, что приложение выполняет все заявленные функции, работает стабильно и корректно обрабатывает входные данные. Ошибок и критических уязвимостей не обнаружено. Приложение готово к передаче заказчику и внедрению в эксплуатацию.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

- 1 Bitrix24 [Электронный ресурс]. – URL: <https://www.bitrix24.ru/>. (дата обращения 14.05.2025).
- 2 Киреев Н. В. Архитектурные компоненты и шаблоны проектирования в разработке программного обеспечения мобильных систем : учебное пособие. Красноярск : СибГУ им. академика М. Ф. Решетнёва, 2022. 90 с.
- 3 Белов А. В. DevOps: принципы, инструменты, внедрение. Санкт-Петербург : Питер, 2023. 288 с.
- 4 Федоров А. Г. Современные веб-технологии: Django REST Framework. Санкт-Петербург : БХВ-Петербург, 2022. 320 с.
- 5 Иванова Л. М. UX/UI-дизайн: современные тренды и кейсы. Екатеринбург : УрФУ, 2023. 176 с.
- 6 Габриелян Т. О. Коммуникативный и мультимедийный дизайн. Графический пользовательский интерфейс : учебно-методическое. Симферополь : КФУ им. В.И. Вернадского, 2021. 166 с.
- 7 Марков В. С. Проектирование баз данных: от концепции к реализации. Москва : ИНФРА-М, 2023. 278 с.
- 8 Нестеров С. А. Базы данных: учебник и практикум для вузов. Москва: Изд-во Юрайт, 2023. 259 с.
- 9 Кара-Ушанов В. Ю. SQL – язык реляционных баз данных: учебное пособие. Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2016. 156 с.
- 10 Петросян Л. Э. Разработка мобильных приложений на языке Kotlin : учебное пособие для вузов. Санкт-Петербург : Лань, 2025. 104 с.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Басыня, Е. А. Сетевая информационная безопасность: учебник / Е. А. Басыня. – Москва : НИЯУ МИФИ, 2023. – 224 с.
- 2 Белов, А. В. DevOps: принципы, инструменты, внедрение / А. В. Белов. – Санкт-Петербург : Питер, 2023. – 288 с.
- 3 Габриелян, Т. О. Коммуникативный и мультимедийный дизайн. Графический пользовательский интерфейс : учебно-методическое пособие / Т. О. Габриелян. – Симферополь : КФУ им. В.И. Вернадского, 2021. – 166 с.
- 4 Григорьев, И. С. Машинное обучение для анализа данных / И. С. Григорьев. – Новосибирск : НГТУ, 2023. – 198 с.
- 5 Горбунов, И.В. Kotlin для Android: профессиональная разработка приложений – СПб.: Питер, 2022. – 412 с.
- 6 Гусев, Д. В. Защита данных в мобильных приложениях / Д. В. Гусев. – Москва : Горячая линия-Телеком, 2023. – 210 с.
- 7 Иванова, Л. М. UX/UI-дизайн: современные тренды и кейсы / Л. М. Иванова. – Екатеринбург : УрФУ, 2023. – 176 с.
- 8 Использование диаграммы вариантов использования UML при проектировании программного обеспечения [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/566218/>. – 25.03.2025.
- 9 Кара-Ушанов, В. Ю. SQL – язык реляционных баз данных: учебное пособие / В. Ю. Кара-Ушанов. – Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2016. – 156 с.
- 10 Киреев, Н. В. Архитектурные компоненты и шаблоны проектирования в разработке программного обеспечения мобильных систем : учебное пособие / Н. В. Киреев. – Красноярск : СибГУ им. академика М. Ф. Решетнёва, 2022. – 90 с.
- 11 Козлов, П. Р. Тестирование программного обеспечения: методы и инструменты / П. Р. Козлов. – Москва : ДМК Пресс, 2021. – 320 с.

12 Марков, В. С. Проектирование баз данных: от концепции к реализации / В. С. Марков. – Москва : ИНФРА-М, 2023. – 278 с.

13 Методические указания к выполнению и защите выпускной квалификационной работы бакалавра / [ред. С.Г. Самохвалова]. – Б.: ФГБОУ ВО «АмГУ», 2025. – 57 с.

14 Нестеров, С. А. Базы данных: учебник и практикум для вузов / С. А. Нестеров. – Москва: Изд-во Юрайт, 2023. – 259 с.

15 Персиваль, Г. Python. Разработка на основе тестирования. Повинуйся Билли-тестировщику, используя Django, Selenium и JavaScript / Г. Персиваль ; перевод с английского А. В. Логунов. – Москва : ДМК Пресс, 2018. – 622 с.

16 Петросян, Л. Э. Разработка мобильных приложений на Kotlin : учебное пособие / Л. Э. Петросян, Н. А. Приходько. – Москва : РТУ МИРЭА, 2024. – 101 с.

17 Петросян, Л. Э. Разработка мобильных приложений на языке Kotlin : учебное пособие для вузов / Л. Э. Петросян, К. В. Гусев. – 2-е изд., стер. – Санкт-Петербург : Лань, 2025. – 104 с.

18 Python 3.12.4 documentation [Электронный ресурс]. – Режим доступа: <https://docs.python.org/>. – 12.04.2025.

19 Раченко, Т. А. Информационная безопасность : учебно-методическое пособие / Т. А. Раченко. – Тольятти : ТГУ, 2024. – 135 с.

20 Смирнов, К. А. Интеграционные решения для CRM-систем / К. А. Смирнов. – Москва : Альфа-Пресс, 2022. – 185 с.

21 Стружкин, Н. П. Базы данных: проектирование: учебник для вузов / Н. П. Стружкин, В. В. Годин. – Москва: Юрайт, 2023. – 477 с.

22 Федоров, А. Г. Современные веб-технологии: Django REST Framework / А. Г. Федоров. – Санкт-Петербург : БХВ-Петербург, 2022. – 320 с.

23 Волков, Д. Н. Кибербезопасность: атаки и защита / Д. Н. Волков. – Москва : Лаборатория знаний, 2022. – 420 с.

24 Bitrix24 [Электронный ресурс]. – Режим доступа: <https://www.bitrix24.ru/>. – 14.05.2025.

25 Bitrix24 REST API [Электронный ресурс]. – Режим
доступа: <https://apidocs.bitrix24.ru/>. – 14.05.2025.

ПРИЛОЖЕНИЕ А

Техническое задание согласно ГОСТ 19.20178

1 Введение

1.1 Наименованием программы

Наименование программы «Android-приложение «Продажа запчастей»».

1.2 Краткая характеристика области применения программы

Данное программное средство предназначено организации процесса продажи запчастей посредством мобильного приложения для операционной системы Android.

2 Основание для разработки

2.1. Основание для проведения разработки

Основанием для разработки является внутренний приказ компании.

2.2. Наименование и условное обозначение темы разработки

Наименование темы разработки – «Разработка Android-приложения «Продажа запчастей» для ООО «УникОпт».

Условное обозначение разработки – «Android-приложение «Продажа запчастей»».

3 Назначение разработки

3.1 Функциональное назначение программы

Программа «Продажа запчастей» предназначена для автоматизации процесса продажи автомобильных запчастей в компании ООО «УникОпт». Её основная функция заключается в предоставлении клиентам удобного инструмента для поиска, выбора и заказа необходимых деталей через мобильное приложение. Для сотрудников программа упрощает процесс обработки заказов, интеграции с CRM-системой Bitrix24.

Продолжение ПРИЛОЖЕНИЯ А

3.2 Эксплуатационное назначение программы

Приложение должно представлять собой удобный и функциональный инструмент, предназначенный для клиентов предприятия, позволяя пользователям самостоятельно формировать заказы, быстро находить нужные запчасти в каталоге, получать актуальную информацию о товарах (включая цены, наличие и характеристики), а также отслеживать статус своих заказов в режиме реального времени.

4 Требования к программе

4.1 Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

Программа должна обеспечивать возможность выполнения следующих функций:

- безопасность и управление доступом, разграничение прав пользователей, защита данных (шифрование, хеширование паролей);
- добавление, удаление товара из корзины, формирование заказов;
- оформление заказов с выбором адреса, способа оплаты и доставки;
- авторизация и регистрация пользователей, личный кабинет клиента;
- каталог товаров с информацией о товаре: наличие товара, его цена, описание;
- сохранение адресов и способов доставки;
- предварительный просчет заказов;
- личный кабинет юридического лица;
- система уведомлений о смене статуса заказа клиента;
- сохранение данных о машинах клиента для персональных подборок запчастей;
- административный раздел с функцией редактирования каталога и обновления остатков и цен;
- синхронизация заказов с CRM Bitrix24.

Продолжение ПРИЛОЖЕНИЯ А

4.1.2 Требования к организации входных данных

Входные данные должны предоставляться в виде структурированных таблиц (CSV, XLSX) или через формы ввода в интерфейсе программы.

Обязательная проверка на дублирование артикулов и наименований, валидация числовых значений, контроль форматов дат и времени. Программа должна проверять входные данные на соответствие формату

Источники входных данных:

- ручной ввод через пользовательский интерфейс;
- ручной ввод через интерфейс административного раздела;
- загрузка из файлов (Excel, CSV).

4.1.3. Требования к временным характеристикам

Обработка пользовательских запросов с использованием серверной части приложения не должна занимать более 3-х секунд реального времени.

Обработка действий пользователя без запросов к серверной части приложения не должна занимать более 2-х секунд реального времени.

4.2 Требования к надёжности

4.2.1. Требования к обеспечению надежного функционирования программы

Программа должна обеспечивать бесперебойную работу при:

- круглосуточной эксплуатации;
- одновременной работе 200 и более пользователей;
- пиковых нагрузках (учет 100+ операций/мин).

Необходима защита от сбоев: автоматическое сохранение данных, обработка исключений, резервное копирование.

Продолжение ПРИЛОЖЕНИЯ А

4.2.2. Время восстановления после отказа

Максимальное время восстановления:

- при программных сбоях: до 15 минут;
- при аппаратных отказах: до 2 часов;
- Полное восстановление из резервной копии: до 4 часов.

Максимальное время восстановления после сбоя работы сервера не должна превышать время восстановления его корректной работы.

4.3. Условия эксплуатации

4.3.1. Климатические условия эксплуатации

Климатические условия эксплуатации определяются, требованиями к климатическим условиям эксплуатации используемых технических средств.

4.3.2. Требования к видам обслуживания

Плановое:

- ежемесячное обновление ПО;
- проверка целостности данных (раз в неделю).

Аварийное:

- устранение ошибок в течение 24 часов;
- восстановление данных при сбоях.

4.3.3. Требования к численности и квалификации персонала

Для развертывания и поддержания работоспособности программного средства достаточно одно технического специалиста со знаниями MySQL, Windows, серверных систем на ОС Linux и базовыми знаниями ОС Android.

Продолжение ПРИЛОЖЕНИЯ А

4.4. Требования к составу и параметрам технических средств

Для функционирования модуля требуется сервер, хостинг с установленным веб-сервером или VDS сервер.

Минимальные требования к техническим средствам сервера:

- процессор с тактовой частотой не ниже 2 ГГц;
- оперативная память от 2 ГБ;
- свободное дисковое пространство от 10 ГБ;
- установленная и настроенная СУБД (MySQL);
- интерпретатор Python версии 3.11 или выше.

4.5. Требования к информационной и программной совместимости

4.5.1. Требования к информационным структурам и методам решения

Программное средство должно использовать реляционную структуру базы данных MySQL с нормализацией до третьей нормальной формы. Для обеспечения производительности требуется создание индексов по ключевым полям. Для обеспечения целостности данных необходимо включить логирование изменений записей в базе данных и резервное копирование.

В качестве методов обработки информации применяются: транзакционный механизм для критически важных операций. Все информационные структуры должны проектироваться с учетом дальнейшего расширения функциональности без необходимости кардинального изменения схемы данных.

Данные между клиентской и серверной частью предписывается передавать в формате JSON для сложных и/или требующих защиты типов данных, в параметрах URL или строки запроса для остальных видов данных, посредством защищенных HTTPS-запросов. Ключи доступа и токены аутентификации должны передаваться только в заголовке запроса, их корректность должна проверяться до выполнения программного кода запроса.

Продолжение ПРИЛОЖЕНИЯ А

4.5.2. Требования к исходным кодам и языкам программирования

Для создания базы данных должен использоваться язык SQL. Разработка клиентской части должна происходить с использованием языков программирования Kotlin и Java. Java необходимо использовать только в случае если это оптимизирует программный код с точки зрения его объема и времени выполнения. Для разработки серверной части предписывается использовать языки программирования Python версии 3.11 и выше и HTML.

Исходные коды программы должны быть документированы и соответствовать основным принципам объектно-ориентированного программирования.

4.5.3. Требования к программным средствам, используемым программой

В качестве системы управления базой данных предписывается использовать «MySQLServer» и язык программирования SQL.

В качестве программного обеспечения сервера необходимо использовать панель управления «ipmanager» с установленным интерпретатором Python и защитой от DDOS-атак.

Клиентскую часть необходимо реализовать с использованием фреймворка Jetpack Compose и библиотек, являющихся его частью.

4.6 Требование к маркировке и упаковке

Требования к упаковке отсутствуют, ПО распространяется по сети «Интернет» или через физические носители данных.

4.7 Требования к транспортированию и хранению

Требования к транспортированию и хранению отсутствуют, ПО распространяется по сети «Интернет» или через физические носители данных.

Продолжение ПРИЛОЖЕНИЯ А

4.8 Специальные требования

4.8.1 Требования к эргономике и технической эстетике

Интерфейс программы требует единого стиля оформления всех модулей. Необходимо минимизация действий пользователя при выполнении типовых задач. Интерфейс должен быть адаптивным, поддерживать различные разрешения экрана и использовать стандартные элементы управления ОС. Техническая эстетика предполагает минималистичный дизайн без избыточных элементов, логичную группировку компонентов и единые стили оформления экранов приложения.

5 Требования к программной документации

Предварительный состав программной документации:

- техническое задание;
- программа и методика испытаний;
- руководство системного программиста;
- руководство пользователя;
- руководство программиста.

Требования к документации устанавливаются следующими документами:

- виды программных документов ГОСТ 19.10177;
- стадии разработки ГОСТ 19.10277;
- техническое задание. Требования к содержанию и оформлению ГОСТ 19.20178;
- схемы алгоритмов, программ данных и систем ГОСТ 19.70190;
- текст программы ГОСТ 19.40178;
- описание программы ГОСТ 19.402 78;
- пояснительная записка ГОСТ 19.40479;
- описание применения ГОСТ 19.50278;

Продолжение ПРИЛОЖЕНИЯ А

- руководство системному программисту ГОСТ 19.50379;
- руководство программиста ГОСТ 19.50479;
- руководство оператору ГОСТ 19.50579.

6 Технико-экономические показатели

Разработка и внедрение приложения «Продажа Запчастей» должно позволить организации автоматизировать основные бизнес-процессы, такие как формирование и отслеживание заказов клиентами. В долгосрочной перспективе программное обеспечение должно увеличить расходы компании и снизить издержки при относительно небольших расходах на поддержание работоспособности ПО.

7 Стадии и этапы разработки

Таблица А.1 – Стадии и этапы разработки

Стадия разработки	Этапы разработки	Сроки	Содержание работ
Техническое задание	Определение требований к программному средству.	9.09.2024 – 19.09.2024	Сбор исходных материалов, определение требований к техническим средствам.
	Разработка технического задания.	20.09.2024 – 10.10.2024	Определение стадий и сроков разработки ПО, определение требований, выбор языков программирования.
Эскизный проект	Разработка эскизного проекта.	11.10.2024 – 30.10.2024	Предварительная разработка структуры данных, общего описания алгоритма решения.
Технический проект	Разработка технического проекта.	01.11.2024 – 12.12.2024	Уточнение структуры данных, стека технологий и разработка структуры программы.
Рабочий проект	Разработка рабочего проекта	13.12.2024 – 30.04.2025	Программирование и отладка программы.
	Разработка программной документации	01.05.2025 – 15.05.2025	Разработка программных документов в соответствии с ГОСТ 19.10177.
Внедрение	Подготовка и передача программы.	16.05.2025 – 01.06.2025	Подготовка и передача программы и документации для сопровождения.

Продолжение ПРИЛОЖЕНИЯ А

8 Порядок контроля и приёмки

Приемка включает предварительные, приемосдаточные и эксплуатационные испытания с проверкой соответствия ТЗ на контрольных примерах. Критерии приемки: 100% выполнение функциональных требований, соответствие показателям надежности и полный комплект документации.

ПРИЛОЖЕНИЕ Б

Логическая модель данных

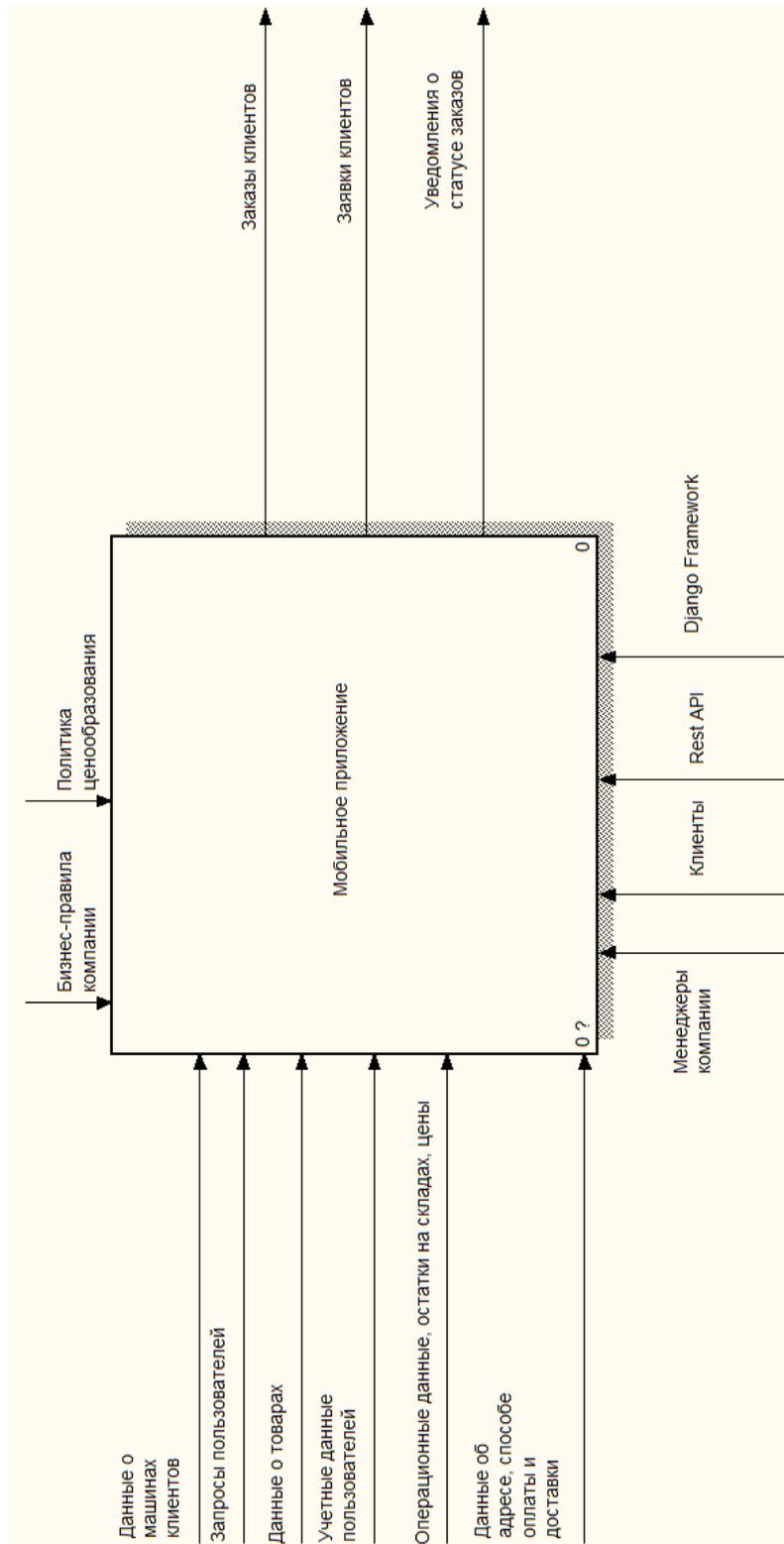


Рисунок Б.1 – Диаграмма IDEF0 нулевого уровня

Продолжение ПРИЛОЖЕНИЯ Б

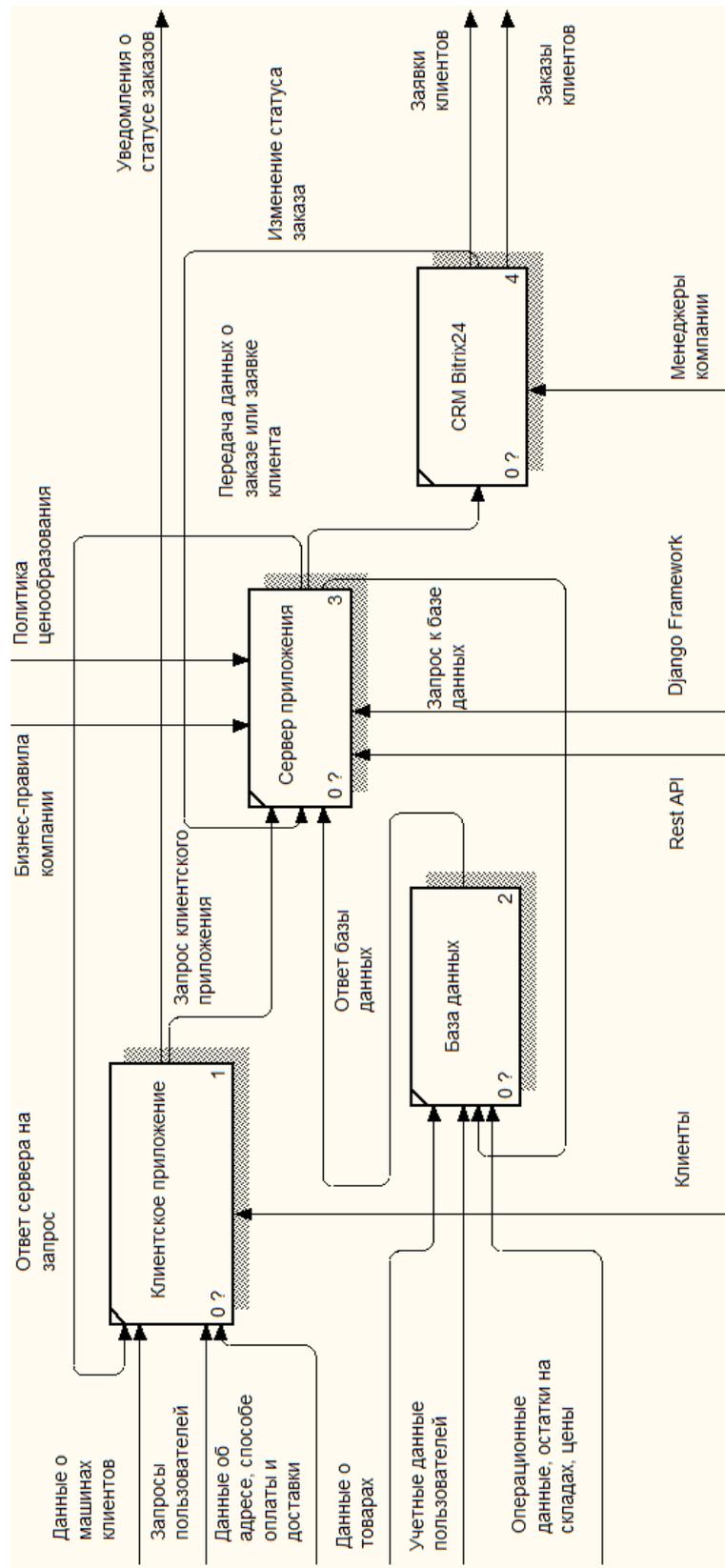


Рисунок Б.2 – Диаграмма IDEF0 первого уровня

ПРИЛОЖЕНИЕ В

Проектирование программного обеспечения

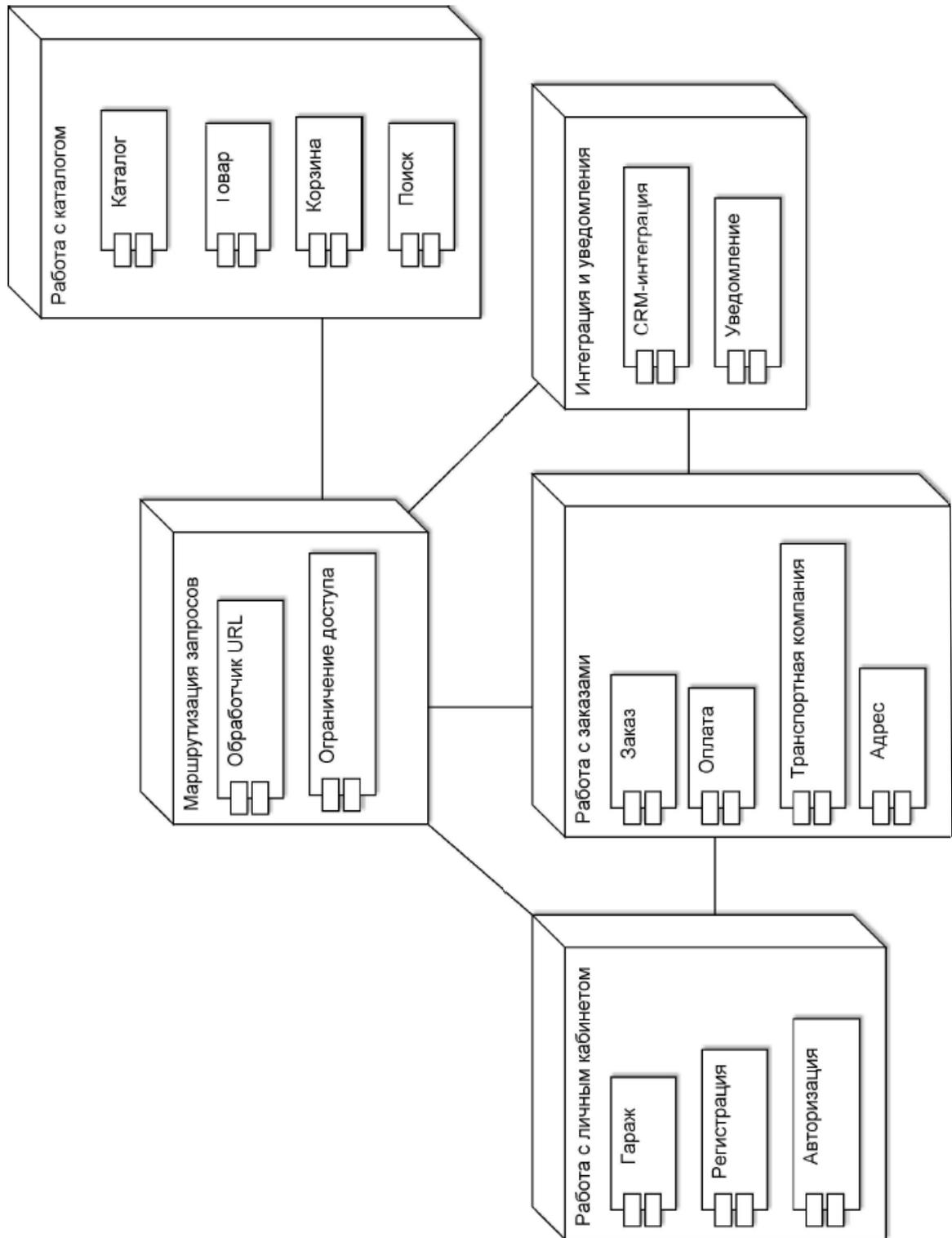


Рисунок В.1 – UML-диаграмма развертывания модулей серверной части

Продолжение ПРИЛОЖЕНИЯ В

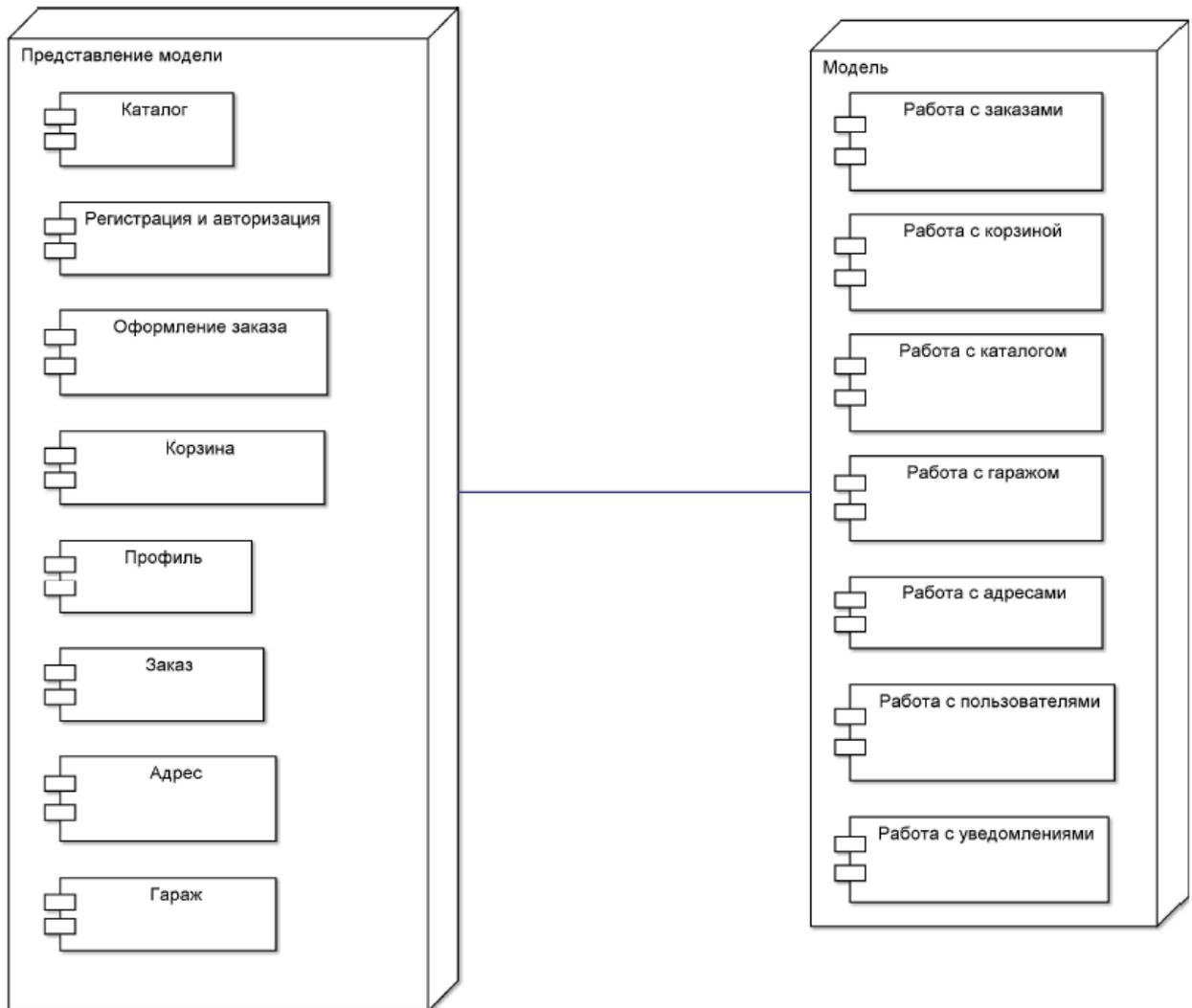


Рисунок В.2 – UML-диаграмма развертывания модулей клиентской части

Продолжение ПРИЛОЖЕНИЯ В



Рисунок В.3 – Прототип каталога приложения

Продолжение ПРИЛОЖЕНИЯ В

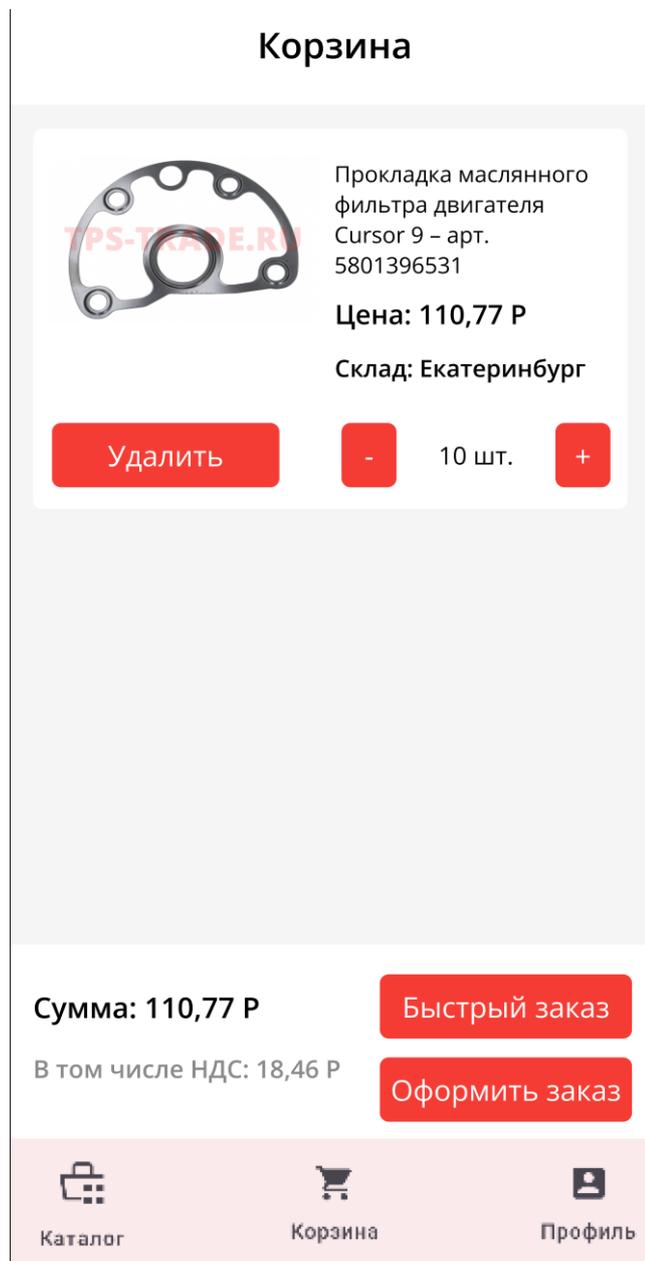


Рисунок В.4 – Прототип корзины приложения

Продолжение ПРИЛОЖЕНИЯ В

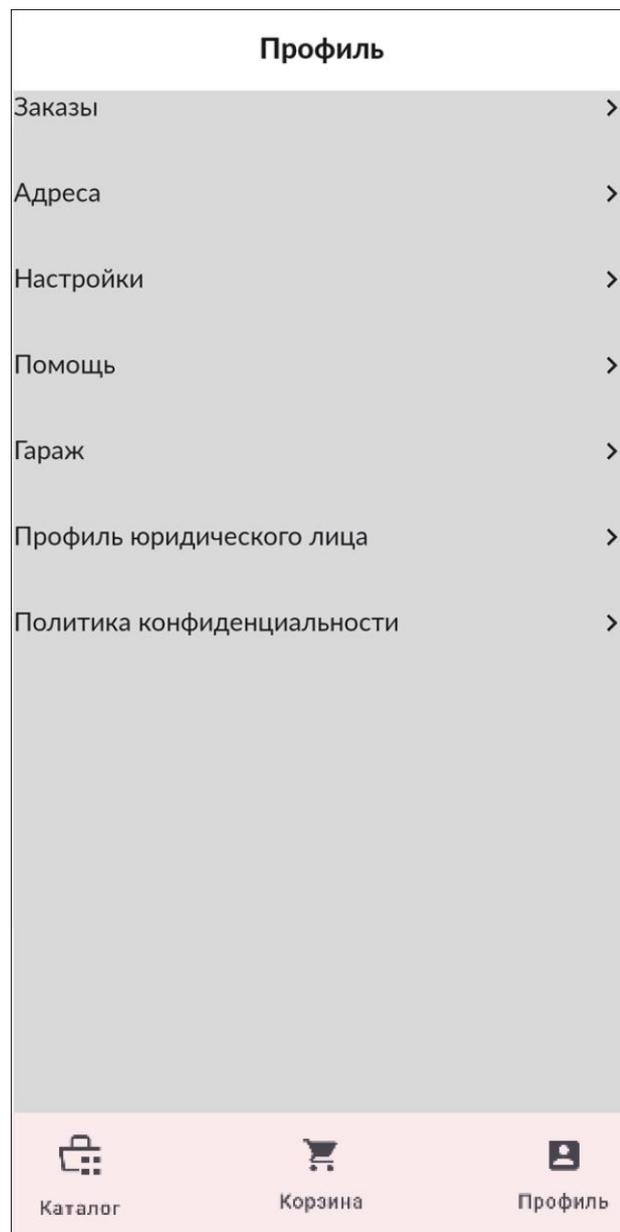


Рисунок В.5 – Прототип личного кабинета приложения

Продолжение ПРИЛОЖЕНИЯ В

Таблица В.1 – Спецификация атрибутов сущности «Пользователь»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код пользователя	Числовой	>0	1, 2, 3
Пароль	Хэш пароля пользователя	Текстовый	–	pbkdf2
Администратор	Является ли пользователь администратором	Логический	true, false	true
Логин	Логин пользователя, используется электронная почта	Текстовый	–	nikita1.log2002@mail.ru
Имя	Имя пользователя	Текстовый	–	Никита
Фамилия	Фамилия пользователя	Текстовый	–	Логинов
Электронная почта	Электронная почта пользователя	Текстовый	–	nikita1.log2002@mail.ru
Активность	Является ли аккаунт пользователя активным	Логический	true, false	true
Телефон	Мобильный телефон пользователя	Текстовый	–	+79145505850

Таблица В.2 – Спецификация атрибутов сущности «Адрес»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код адреса	Числовой	>0	1, 2, 3
Регион	Регион доставки	Текстовый	–	Амурская область
Город	Город доставки	Текстовый	–	Благовещенск
Точный адрес	Строка для ввода улицы, номера дома, квартиры	Текстовый	–	Ленина, 1, квартира 43
Активность	Активность адреса	Логический	true, false	true

Продолжение ПРИЛОЖЕНИЯ В

Таблица В.3 – Спецификация атрибутов сущности «Ключ доступа»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код ключа	Числовой	>0	1, 2, 3
Ключ	Ключ доступа к серверу	Текстовый	–	MNibEKAsXewM

Таблица В.4 – Спецификация атрибутов сущности «Содержимое корзины»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код записи	Числовой	>0	1, 2, 3
Количество	Количество товара в корзине	Числовой	>0	1, 2, 3

Таблица В.5 – Спецификация атрибутов сущности «Заказ»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код заказа	Числовой	>0	1, 2, 3
Сумма	Сумма заказа	Числовой	>0	150,0
Дата	Дата заказа	Дата	=>Текущая дата	25.10.2025
Оплата	Оплачен ли заказ	Логический	true, false	true
Код лида	Код лида CRM битрикс	Числовой	>0	1, 2, 3
Синхронизация	Синхронизирован ли заказ с CRM	Логический	true, false	true

Таблица В.6 – Спецификация атрибутов сущности «Происхождение товара»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код происхождения	Числовой	>0	1, 2, 3
Наименование	Наименование происхождения товара	Текстовый	–	Оригинал

Продолжение ПРИЛОЖЕНИЯ В

Таблица В.7 – Спецификация атрибутов сущности «Тип цены»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код типа цены	Числовой	>0	1, 2, 3
Наименование	Наименование типа цены	Текстовый	–	Оптовая цена

Таблица В.8 – Спецификация атрибутов сущности «Цена на товар»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код цены	Числовой	>0	1, 2, 3
Цена	Цена на товар	Числовой	>0	150,2

Таблица В.9 – Спецификация атрибутов сущности «Картинка»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код картинки	Числовой	>0	1, 2, 3
Путь	Путь к картинке	Текстовый	–	/pictures/path.webp
Путь миниатюры	Путь к уменьшенной версии картинки	Текстовый	–	/pictures/path_t.webp
Главная картинка	Является ли фото главным изображением товара	Логический	true, false	true

Таблица В.10 – Спецификация атрибутов сущности «Каталог»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код товара	Числовой	>0	1, 2, 3
Артикул	Артикул товара	Текстовый	–	5801254056
Наименование	Наименование товара	Текстовый	–	Подшипник подвесной
Описание	Описание товара	Текстовый	–	Подшипник для машин

Продолжение ПРИЛОЖЕНИЯ В

Продолжение таблицы В.1

Активность	Активен ли товар	Логический	true, false	true
------------	------------------	------------	-------------	------

Таблица В.11 – Спецификация атрибутов сущности «Производитель»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код производителя	Числовой	>0	1, 2, 3
Наименование	Наименование производителя	Текстовый	–	Bosch

Таблица В.12 – Спецификация атрибутов сущности «Способ оплаты»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код способа оплаты	Числовой	>0	1, 2, 3
Наименование	Наименование способа оплаты	Текстовый	–	Bosch

Таблица В.13 – Спецификация атрибутов сущности «Транспортная компания»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код транспортной компании	Числовой	>0	1, 2, 3
Наименование	Наименование транспортной компании	Текстовый	–	Деловые линии
Описание	Описание транспортной компании	Текстовый	–	Бесплатная доставка до терминала деловые линии

Таблица В.14 – Спецификация атрибутов сущности «Статус заказа»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код статуса заказа	Числовой	>0	1, 2, 3

Продолжение ПРИЛОЖЕНИЯ В

Продолжение таблицы В.14

Наименование	Наименование статуса заказа	Текстовый	–	Отправлен
--------------	-----------------------------	-----------	---	-----------

Таблица В.15 – Спецификация атрибутов сущности «Склад»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код склада	Числовой	>0	1, 2, 3
Наименование	Наименование склада	Текстовый	–	Склад Благовещенск
Адрес	Адрес склада	Текстовый	–	Текстильная, 112/1

Таблица В.16 – Спецификация атрибутов сущности «Юридическое лицо»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код юридического лица	Числовой	>0	1, 2, 3
Наименование	Наименование юридического лица	Текстовый	–	Склад Благовещенск
ИНН	ИНН юридического лица	Текстовый	–	7108700925
КПП	КПП юридического лица	Текстовый	–	770201001
Адрес	Адрес юридического лица	Текстовый	–	Амурская область, Благовещенск
Расчетный счет	Номер расчетного счета	Текстовый	–	40702855809560000381
БИК	БИК банка расчетного счета	Текстовый	–	044525411

Таблица В.17 – Спецификация атрибутов сущности «Содержание заказа»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код содержания	Числовой	>0	1, 2, 3

Продолжение ПРИЛОЖЕНИЯ В

Продолжение таблицы В.17

Цена	Цена товара	Числовой	>0	125,5
Количество	Количество товара	Числовой	>0	10

Таблица В.18 – Спецификация атрибутов сущности «Остаток товара»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код остатка	Числовой	>0	1, 2, 3
Количество	Количество товара	Числовой	>0	10

Таблица В.19 – Спецификация атрибутов сущности «Категория товара»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код категории	Числовой	>0	1, 2, 3
Наименование	Наименование категории	Текстовый	–	Запчасти для двигателей
Уровень	Уровень категории	Числовой	>=0	1, 2, 3
Символьный код	Символьный код категории	Текстовый	–	engine_parts
Активность	Активна ли категория	Логический	true, false	true
Код родителя	Код родительской категории	Числовой	>=0	1, 2, 3

Таблица В.20 – Спецификация атрибутов сущности «Токен аутентификации»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Токен	Токен аутентификации пользователя	Текстовый	–	209fabb40411ed
Дата	Дата создания	Текстовый	<=текущая дата	25.12.2024
Код устройства	Уникальный код устройства	Текстовый	–	sd_fabb40411edfds

Продолжение ПРИЛОЖЕНИЯ В

Таблица В.21 – Спецификация атрибутов сущности «Содержимое гаража»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код машины	Числовой	>0	1, 2, 3
ВИН	ВИН-код машины	Текстовый	–	LZFH18PD0012

Таблица В.22 – Спецификация атрибутов сущности «Производитель машины»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код производителя	Числовой	>0	1, 2, 3
Наименование	Наименование производителя	Текстовый	–	Iveco

Таблица В.23 – Спецификация атрибутов сущности «Модель машины»

Название атрибута	Описание	Тип данных	Диапазон значений	Пример
Код	Код модели	Числовой	>0	1, 2, 3
Наименование	Наименование модели	Текстовый	–	CQ3346HV35
Описание	Описание модели	Текстовый	–	350 лошадиных сил, объем кузова 20 кубических метров

ПРИЛОЖЕНИЕ Г

Логическое проектирование базы данных

Сущность «Адрес»

<u>Код</u>
<u>Код пользователя</u>
Регион
Город
Точный адрес
Активность

Сущность «Пользователь»

<u>Код</u>
Пароль
Администратор
Логин
Имя
Фамилия
Электронная почта
Активность
Телефон



Рисунок Г.1 – Отношение «Пользователь» и «Адрес»

Сущность
«Содержимое корзины»

<u>Код</u>
<u>Код пользователя</u>
Количество

Сущность «Пользователь»

<u>Код</u>
Пароль
Администратор
Логин
Имя
Фамилия
Электронная почта
Активность
Телефон



Рисунок Г.2 – Отношение «Содержимое корзины» и «Пользователь»

Продолжение ПРИЛОЖЕНИЯ Г



Рисунок Г.3 – Отношение «Заказ» и «Пользователь»



Рисунок Г.4 – Отношение «Содержимое гаража» и «Пользователь»

Продолжение ПРИЛОЖЕНИЯ Г



Рисунок Г.5 – Отношение «Токен аутентификации» и «Пользователь»

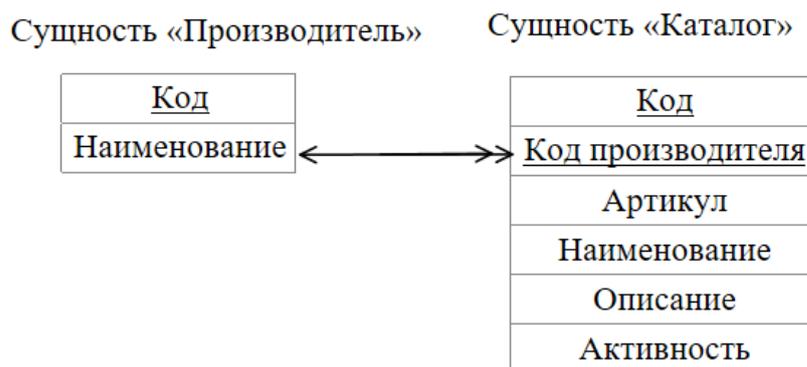


Рисунок Г.6 – Отношение «Производитель» и «Каталог»

Продолжение ПРИЛОЖЕНИЯ Г

Сущность «Категория товара» Сущность «Каталог»

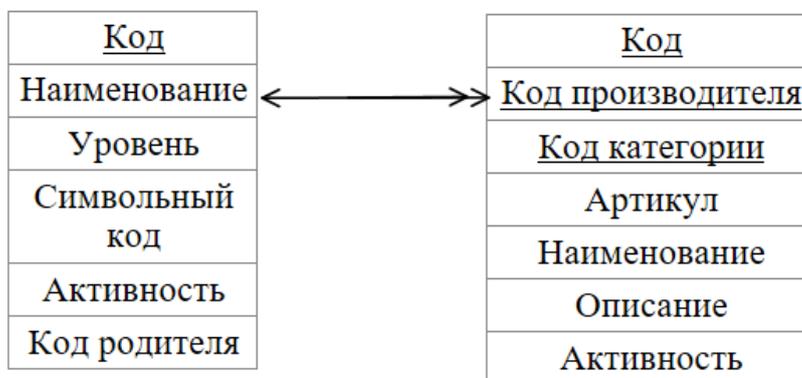


Рисунок Г.7 – Отношение «Категория товара» и «Каталог»

Сущность «Происхождение товара» Сущность «Каталог»



Рисунок Г.8 – Отношение «Происхождение товара» и «Каталог»

Продолжение ПРИЛОЖЕНИЯ Г



Рисунок Г.9 – Отношение «Содержание заказа» и «Каталог»



Рисунок Г.10 – Отношение «Остаток товара» и «Каталог»

Продолжение ПРИЛОЖЕНИЯ Г

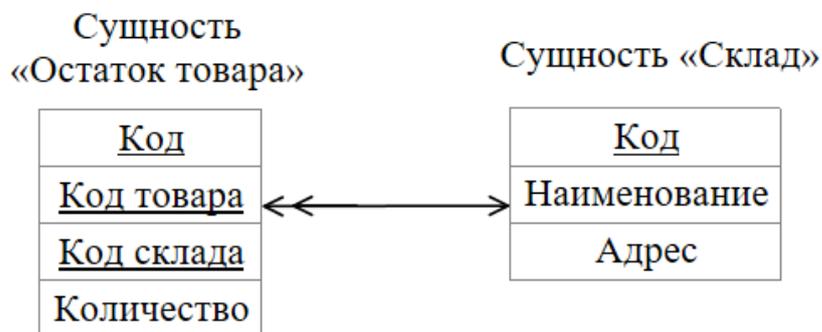


Рисунок Г.11 – Отношение «Остаток товара» и «Склад»

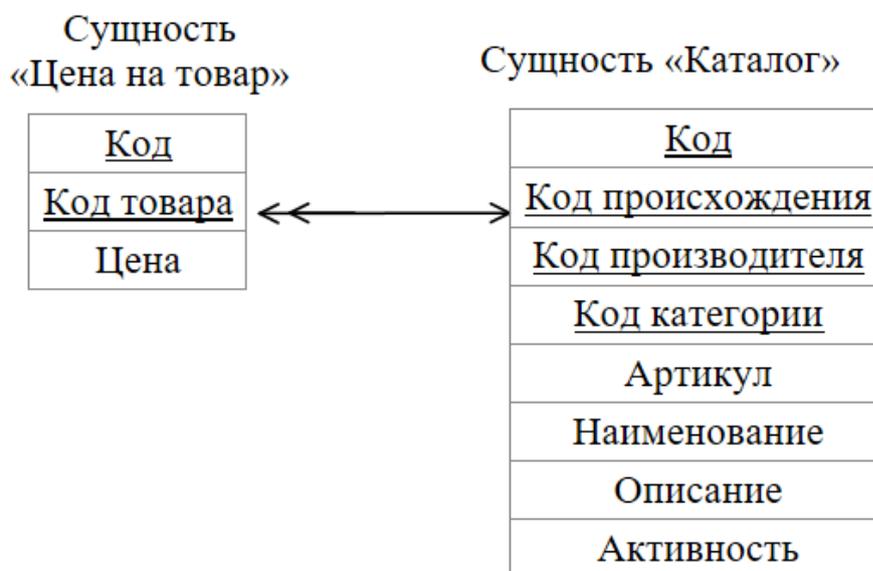


Рисунок Г.12 – Отношение «Цена на товар» и «Каталог»

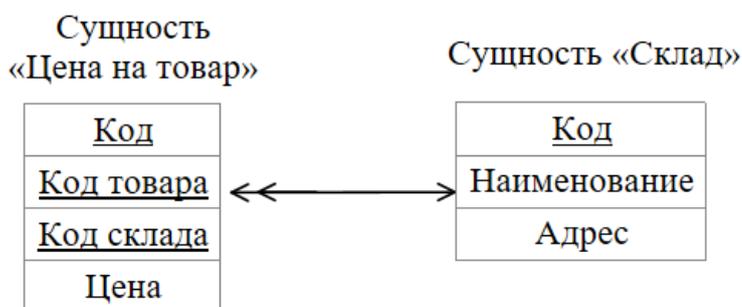


Рисунок Г.13 – Отношение «Цена на товар» и «Склад»

Продолжение ПРИЛОЖЕНИЯ Г

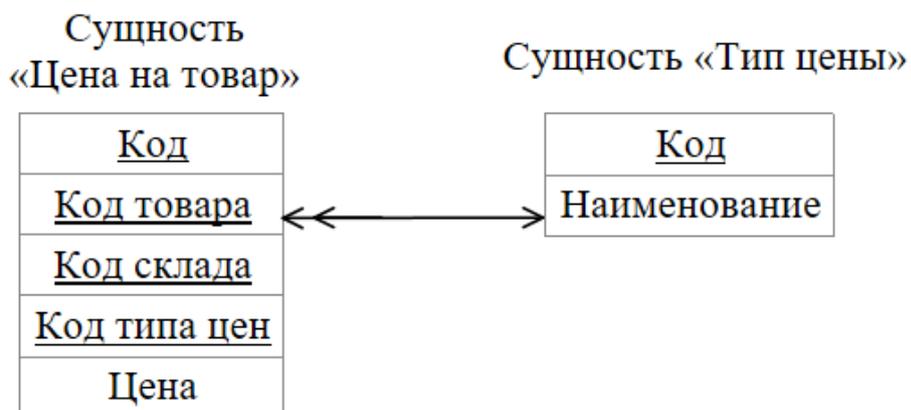


Рисунок Г.14 – Отношение «Цена на товар» и «Тип цены»

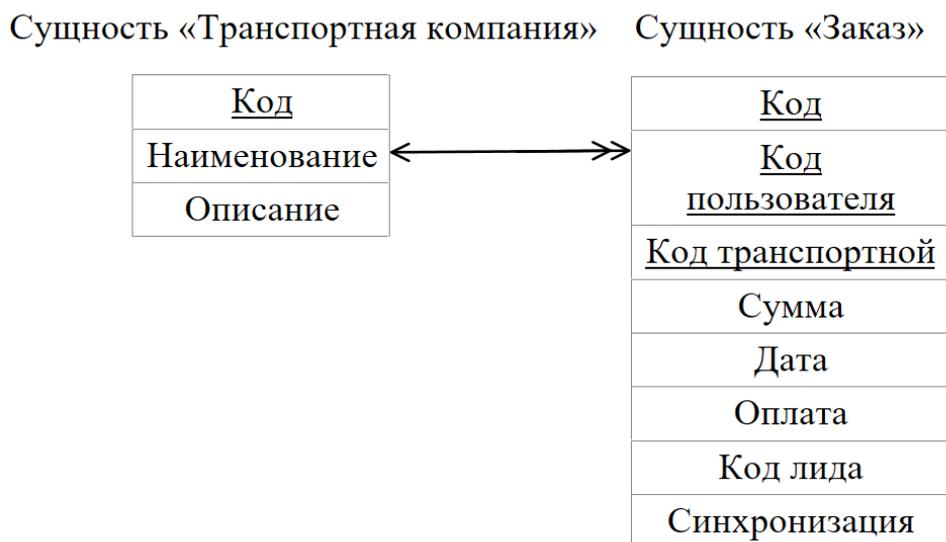


Рисунок Г.15 – Отношение «Заказ» и «Транспортная компания»

Продолжение ПРИЛОЖЕНИЯ Г



Рисунок Г.16 – Отношение «Содержание заказа» и «Заказ»



Рисунок Г.17 – Отношение «Способ оплаты» и «Заказ»

Продолжение ПРИЛОЖЕНИЯ Г



Рисунок Г.18 – Отношение «Адрес» и «Заказ»

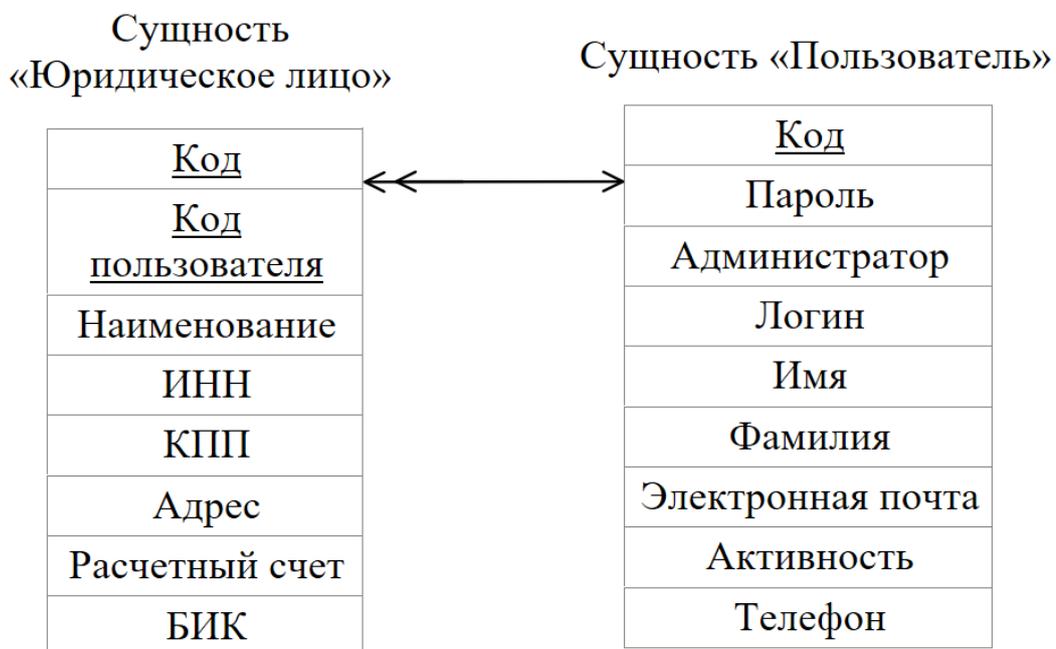


Рисунок Г.19 – Отношение «Юридическое лицо» и «Пользователь»

Продолжение ПРИЛОЖЕНИЯ Г

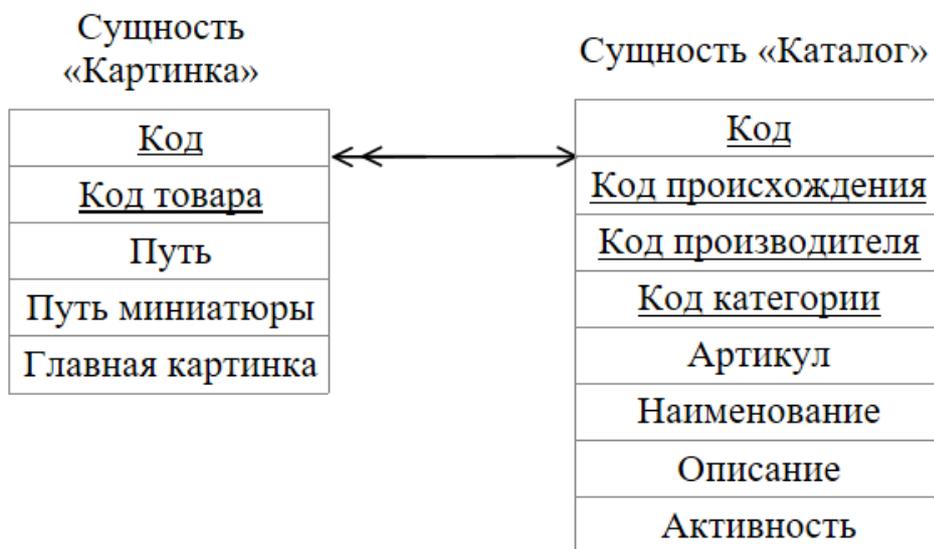


Рисунок Г.20 – Отношение «Картинка» и «Каталог»



Рисунок Г.21 – Отношение «Модель машины» и «Производитель машины»



Рисунок Г.22 – Отношение «Модель машины» и «Содержимое гаража»

Продолжение ПРИЛОЖЕНИЯ Г



Рисунок Г.23 – Отношение «Модель машины» и «Каталог»

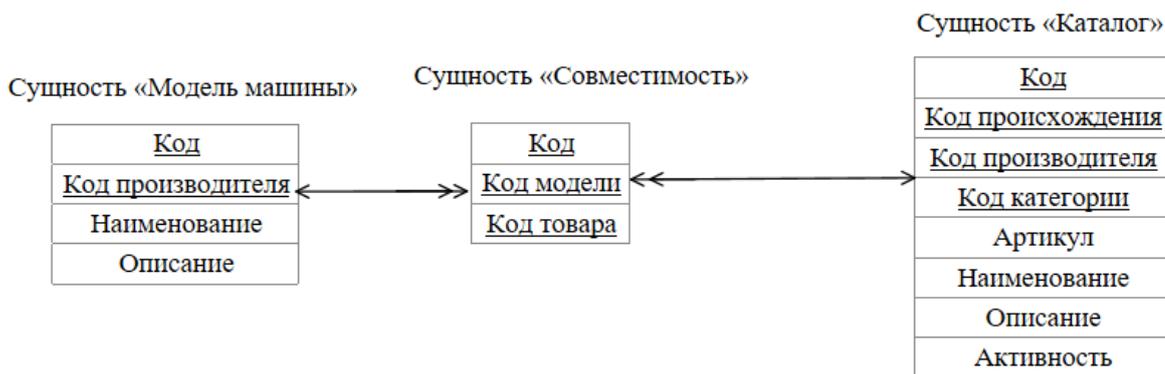


Рисунок Г.24 – Итоговое отношение «Модель машины» и «Каталог»



Рисунок Г.25 – Отношение «Содержимое корзины» и «Каталог»

Продолжение ПРИЛОЖЕНИЯ Г

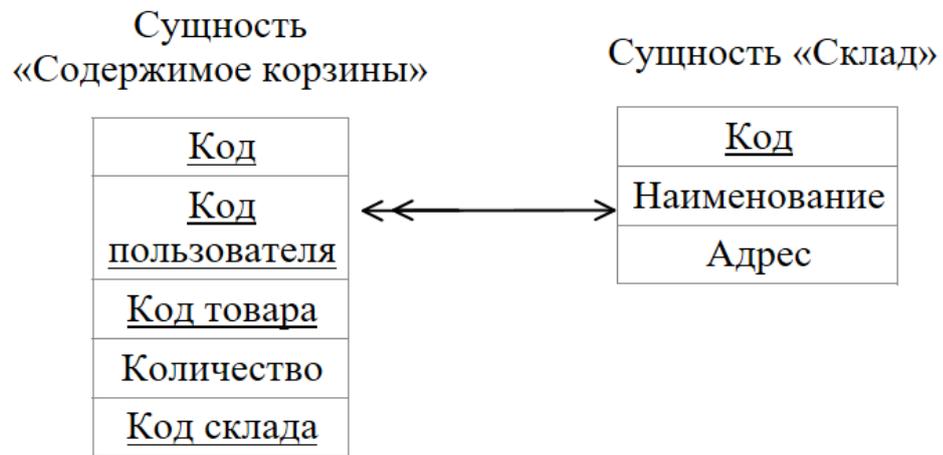


Рисунок Г.26 – Отношение «Содержимое корзины» и «Склад»

Продолжение ПРИЛОЖЕНИЯ Г

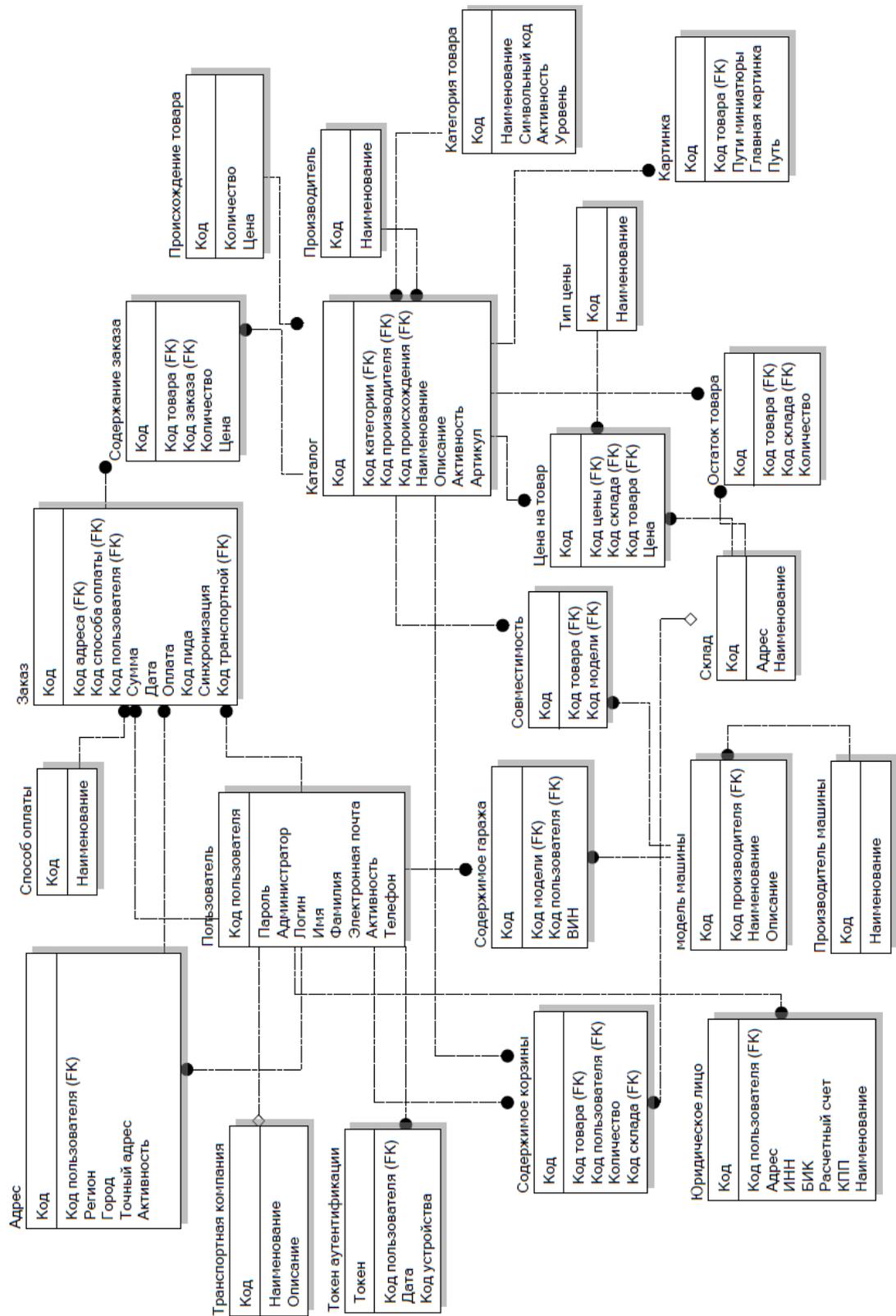


Рисунок Г.27 – Инфологическая модель базы данных

ПРИЛОЖЕНИЕ Д

Физическое проектирование базы данных

Таблица Д.1 – Физическая структура данных сущности «address»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
AddressId	Числовой	>0	int	Primary key
City	Текстовый	–	varchar(255)	–
ExactAddress	Текстовый	–	text	–
Region	Текстовый	–	varchar(255)	–
TransportCompanyId	Текстовый	>0	int	Foreign key
UserId	Числовой	>0	int	Foreign key
IsAcrive	Логический	–	tinyint(1)	–

Таблица Д.2 – Физическая структура данных сущности «api_key»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
KeyId	Числовой	>0	int	Primary key
ApiKey	Текстовый	–	char(60)	–

Таблица Д.3 – Физическая структура данных сущности «auth_user»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
Id	Числовой	>0	int	Primary key
password	Текстовый	–	varchar(128)	–
last_login	Дата	–	datetime(6)	–
is_superuser	Логический	–	tinyint(1)	–
username	Текстовый	–	varchar(150)	–
first_name	Текстовый	–	varchar(150)	–
last_name	Текстовый	–	varchar(150)	–
email	Текстовый	–	varchar(245)	–
phone	Текстовый	–	varchar(200)	–
is_active	Логический	–	tinyint(1)	–
date_joined	Дата	–	datetime(6)	–

Таблица Д.4 – Физическая структура данных сущности «authtoken_token»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
key	Текстовый	–	varchar(40)	Primary key
created	Текстовый	–	datetime(6)	–
user_id	Числовой	>0	int	Foreign key

Продолжение ПРИЛОЖЕНИЯ Д

Таблица Д.5 – Физическая структура данных сущности «car_manufacturer»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
Id	Числовой	>0	int	Primary key
ManufacturerName	Текстовый	–	varchar(255)	–

Таблица Д.6 – Физическая структура данных сущности «car_model»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
ModelId	Числовой	>0	int	Primary key
ModelName	Текстовый	–	varchar(255)	–
ManufacturerId	Числовой	>0	int	Foreign key
Description	Текстовый	–	text	–

Таблица Д.7 – Физическая структура данных сущности «cart»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
Id	Числовой	>0	int	Primary key
UserId	Числовой	>0	int	Foreign key
ProductId	Числовой	>0	int	Foreign key
Quantity	Числовой	>0	int	–
StockId	Числовой	>0	int	Foreign key

Таблица Д.8 – Физическая структура данных сущности «comprability»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
Id	Числовой	>0	int	Primary key
ProductId	Числовой	>0	int	Foreign key
CarId	Числовой	>0	int	Foreign key

Таблица Д.9 – Физическая структура данных сущности «garage»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
Id	Числовой	>0	int	Primary key
ВИН	Текстовый	–	varchar(20)	–
UserId	Числовой	>0	int	Foreign key
ModelId	Числовой	>0	int	Foreign key

Продолжение ПРИЛОЖЕНИЯ Д

Таблица Д.10 – Физическая структура данных сущности «legal_entity»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
Id	Числовой	>0	int	Primary key
UserId	Числовой	>0	int	Foreign key
CompanyName	Числовой	–	varchar(255)	–
Inn	Текстовый	–	varchar(12)	–
Kpp	Текстовый	–	varchar(9)	–
Address	Текстовый	–	text	–
BankAccount	Текстовый	–	varchar(50)	–
Bik	Текстовый	–	varchar(50)	–

Таблица Д.11 – Физическая структура данных сущности «order_content»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
ContentId	Числовой	>0	int	Primary key
OrderId	Числовой	>0	int	Foreign key
ProductId	Числовой	>0	int	Foreign key
Price	Текстовый	–	decimal(10,2)	–
Quantity	Числовой	>0	int	–
StockId	Числовой	>0	int	Foreign key

Таблица Д.12 – Физическая структура данных сущности «order_status»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
StatusId	Числовой	>0	int	Primary key
StatusName	Текстовый	–	varchar(255)	–

Таблица Д.13 – Физическая структура данных сущности «payment_method»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
Id	Числовой	>0	int	Primary key
MethodName	Текстовый	–	varchar(255)	–

Таблица Д.14 – Физическая структура данных сущности «price_type»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
PriceTypeId	Числовой	>0	int	Primary key

Продолжение ПРИЛОЖЕНИЯ Д

Продолжение таблицы Д.14

PriceTypeName	Текстовый	–	varchar(255)	–
---------------	-----------	---	--------------	---

Таблица Д.15 – Физическая структура данных сущности «picture»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
Id	Числовой	>0	int	Primary key
ProductId	Числовой	>0	int	Foreign key
PicturePath	Текстовый	–	varchar(255)	–
PrimaryPicture	Логический	–	tinyint(1)	–
ThumbnailPath	Текстовый	–	varchar(255)	–

Таблица Д.16 – Физическая структура данных сущности «product_catalog»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
ProductId	Числовой	>0	int	Primary key
ProductSku	Текстовый	–	varchar(255)	–
ProductName	Текстовый	–	varchar(255)	–
ProductCategoryId	Числовой	>0	int	Foreign key
Description	Текстовый	–	text	–
OriginId	Числовой	>0	int	Foreign key
IsActive	Логический	–	tinyint(1)	–
ManufacturerId	Числовой	>0	int	Foreign key

Таблица Д.17 – Физическая структура данных сущности «product_category»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
CategoryId	Числовой	>0	int	Primary key
CategoryName	Текстовый	–	varchar(255)	–
CategoryPicturePath	Текстовый	–	varchar(255)	–
CategoryLevel	Числовой	>0	int	–
SymbolCode	Текстовый	–	varchar(255)	–
IsActive	Логический	–	tinyint(1)	–
ParentId	Числовой	>0	int	–

Продолжение ПРИЛОЖЕНИЯ Д

Таблица Д.18 – Физическая структура данных сущности «product_manuf-acturer»

Название атрибута	Тип данных	Ограничен ия	Формат данных	Индексация
ManufacturerId	Числовой	>0	int	Primary key
ManufacturerName	Текстовый	–	varchar(255)	–

Таблица Д.19 – Физическая структура данных сущности «transport_companies»

Название атрибута	Тип данных	Ограничен ия	Формат данных	Индексация
Id	Числовой	>0	int	Primary key
CompanyName	Текстовый	–	varchar(255)	–
CompanyDescription	Текстовый	–	text	–

Таблица Д.20 – Физическая структура данных сущности «stock»

Название атрибута	Тип данных	Ограничен ия	Формат данных	Индексация
StockId	Числовой	>0	int	Primary key
StockName	Текстовый	–	varchar(30)	–
StockAdress	Текстовый	–	varchar(255)	–

Таблица Д.21 – Физическая структура данных сущности «product_quantity»

Название атрибута	Тип данных	Ограничен ия	Формат данных	Индексация
QuantityId	Числовой	>0	int	Primary key
StockId	Числовой	>0	int	Foreign key
ProductId	Числовой	>0	int	Foreign key
Quantity	Числовой	>0	int	–

Таблица Д.22 – Физическая структура данных сущности «product_price»

Название атрибута	Тип данных	Ограничен ия	Формат данных	Индексация
PriceId	Числовой	>0	int	Primary key
Price	Числовой	>0	decimal(10,2)	–
PriceTypeId	Числовой	>0	int	Foreign key
ProductId	Числовой	>0	int	Foreign key
StockId	Числовой	>0	int	Foreign key

Продолжение ПРИЛОЖЕНИЯ Д

Таблица Д.23 – Физическая структура данных сущности «product_origin»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
OriginId	Числовой	>0	int	Primary key
OriginName	Текстовый	–	varchar(255)	–

Таблица Д.24 – Физическая структура данных сущности «product_order»

Название атрибута	Тип данных	Ограничение	Формат данных	Индексация
OrderId	Числовой	>0	int	Primary key
UserId	Числовой	>0	int	Foreign key
OrderStatusId	Числовой	>0	int	Foreign key
OrderDate	Дата	–	datetime	–
OrderSum	Числовой	>0	decimal(10,2)	–
IsPaid	Логический	–	tinyint(1)	–
BitrixLeadCode	Текстовый	–	int	–
PaymentMethodId	Числовой	>0	int	Foreign key
IsSynchronized	Логический	–	tinyint(1)	–
AddressId	Числовой	>0	int	Foreign key
Comment	Текстовый	–	text	–

Продолжение ПРИЛОЖЕНИЯ Д

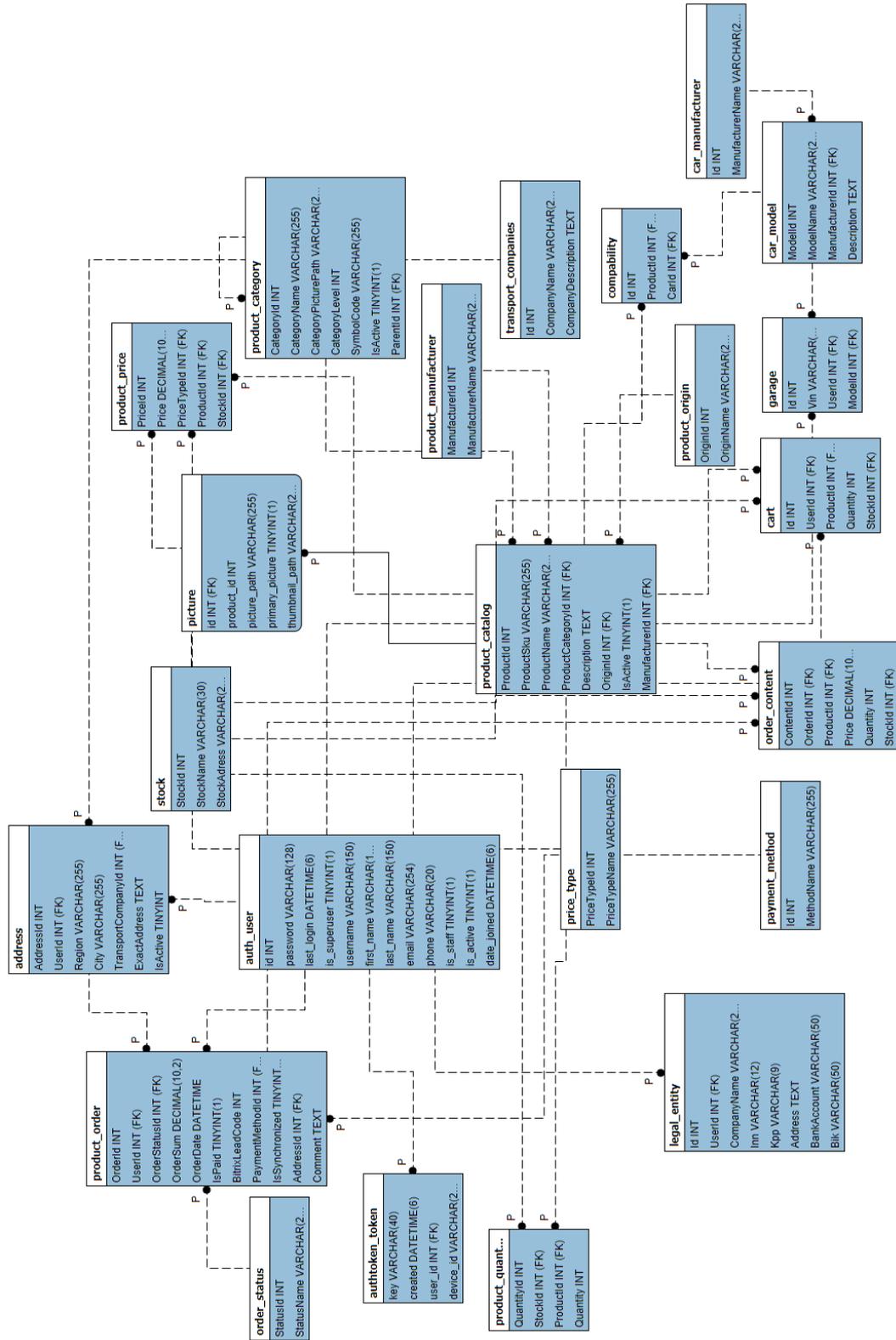


Рисунок Д.1 – Физическая модель базы данных

ПРИЛОЖЕНИЕ Е

Логическая структура

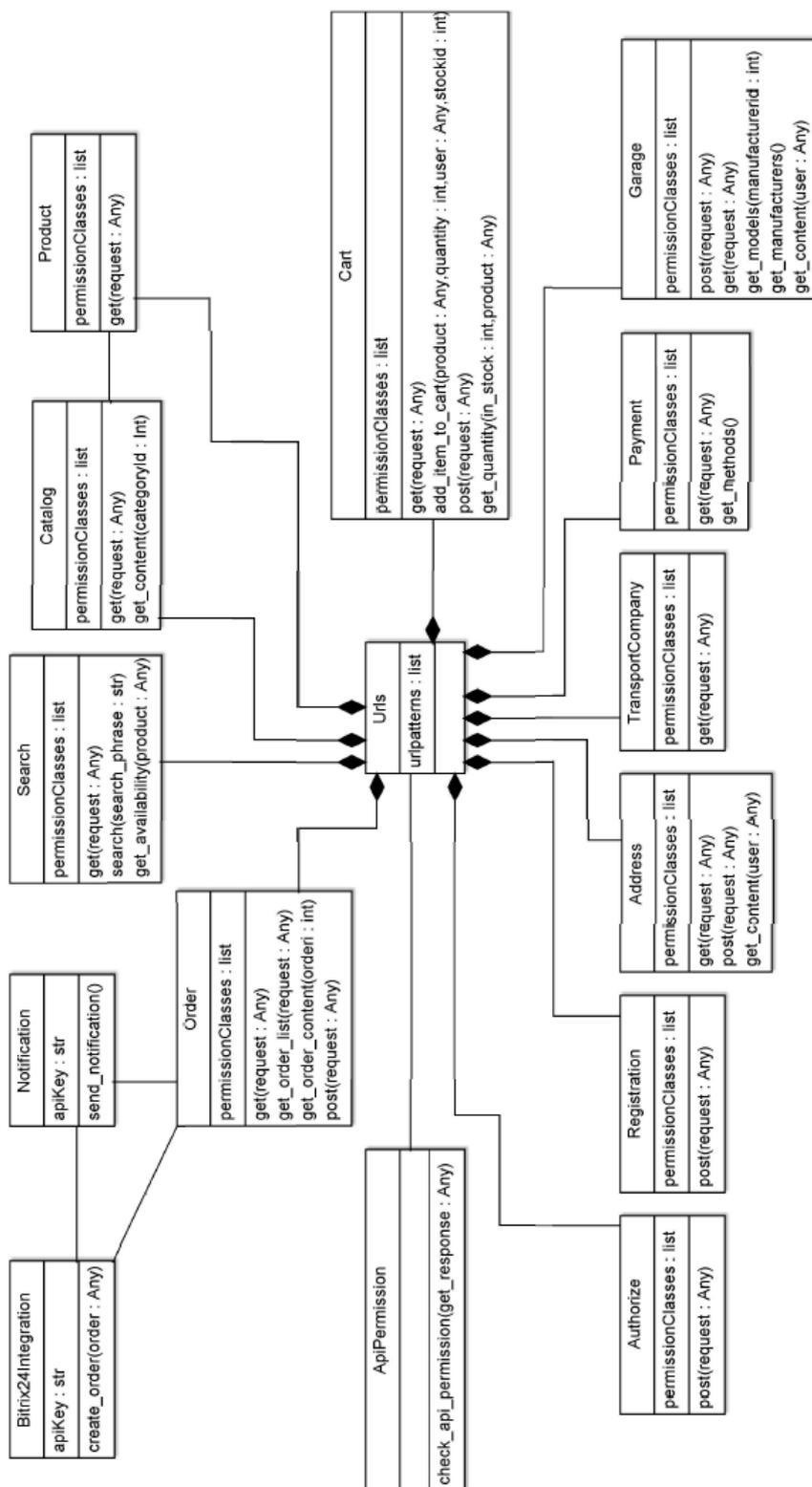


Рисунок Е.1 – UML-диаграмма классов серверной части