

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки, специальности)

Бизнес-информатика
(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка веб-представительства для предприятия оптово-розничной торговли»

Обучающийся

Д.В. Кузьменчук
(Инициалы Фамилия)

(личная подпись)

Руководитель

Т.Г. Любивая

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

Аннотация

Выпускная квалификационная работа выполнена на тему «Разработка веб-представительства для предприятия оптово-розничной торговли».

Объектом исследования является деятельность предприятия оптово-розничной торговли.

Предметом исследования является автоматизация деятельности предприятия оптово-розничной торговли.

Цель выпускной квалификационной работы – разработка веб-представительства для предприятия оптово-розничной торговли.

Бакалаврская работа содержит введение, три главы, заключение, список используемой литературы.

В первой главе проведен анализ деятельности предприятия оптово-розничной торговли, построены функциональные модели бизнес-процессов AS-IS и TO-BE, определены требования к автоматизированному решению.

Во второй главе выполнено описание логической модели веб-представительства, разработана логическая модель данных, сформулированы требования к аппаратному и программному обеспечению.

В третьей главе разработана физическая модель данных, приведен контрольный пример проектного решения, выполнена оценка экономической эффективности проекта.

Оглавление

Введение.....	5
Глава 1 Функциональное моделирование предметной области	7
1.1 Характеристика деятельности предприятия оптово-розничной торговли.....	7
1.2 Разработка функциональной модели бизнес-процессов AS-IS и обоснование необходимости автоматизированного варианта решения	14
1.2.1 Разработка функциональной модели бизнес-процесса управления поставками товара.....	14
1.2.2 Разработка функциональной модели бизнес-процесса продаж.....	15
1.3 Разработка функциональной модели бизнес-процессов TO-BE	18
1.4 Формирование требований к автоматизированному решению	20
1.5 Анализ существующих разработок на предмет соответствия сформулированным требованиям	21
Глава 2 Логическое проектирование	23
2.1 Логическая модель информационной системы и её описание	23
2.1.1 Постановка задачи логического моделирования	23
2.1.2 Компоненты системы и их описание	23
2.1.3 Аппаратные компоненты системы	24
2.1.4 Программные компоненты системы	26
2.2 Разработка логической модели данных.....	28
2.2.1 Сущности базы данных	28
2.2.2. Моделирование процесса наполнения корзины товаров	30
2.3 Информационное и техническое обеспечение	32
Глава 3 Физическое проектирование.....	38
3.1 Системная архитектура проекта	38
3.2 Выбор СУБД и программных средств веб-разработки	41

3.3 Разработка физической модели данных	44
3.3.1 Список таблиц базы данных	44
3.3.2 Таблицы базы данных.....	45
3.4 Контрольный пример реализации проекта	53
3.5 Оценка экономической эффективности проекта.....	59
Заключение.....	66
Список используемой литературы.....	67

Введение

В настоящее время веб-представительство является неотъемлемой частью успешного бизнеса в сфере оптово-розничной торговли. Это связано с тем, что с каждым днем все больше покупателей предпочитают совершать онлайн-покупки, вместо посещения торговых точек в реальном мире. Для предприятий в сфере оптово-розничной торговли веб-представительство является не только способом привлечения новых клиентов, но и инструментом улучшения качества обслуживания уже существующих покупателей.

В данной работе рассмотрена разработка веб-представительства для предприятия оптово-розничной торговли, что позволит предприятию увеличить объемы продаж и улучшить качество обслуживания клиентов.

Объектом исследования является деятельность предприятия оптово-розничной торговли.

Предметом исследования является автоматизация деятельности предприятия оптово-розничной торговли.

Цель выпускной квалификационной работы – разработка веб-представительства для предприятия оптово-розничной торговли.

Для достижения поставленной цели в процессе работы необходимо решить следующие задачи:

- выполнить анализ предметной области;
- построить функциональные модели бизнес-процессов AS-IS и TO-BE;
- сформировать требования к автоматизированному решению;
- разработать логическую и физическую модели данных;
- разработать веб-представительство для предприятия оптово-розничной торговли;
- выполнить оценку экономической эффективности проекта.

Структуру выпускной квалификационной работы составляют введение, основная часть (состоит из трех глав), заключение и список используемой литературы.

Первая глава содержит характеристику деятельности предприятия оптово-розничной торговли, функциональные модели бизнес-процессов AS-IS и TO-BE, требования к автоматизированному решению.

Во второй главе выполнено описание логической модели веб-представительства, разработана логическая модель данных, сформулированы требования к аппаратному и программному обеспечению.

В третьей главе разработана физическая модель данных, приведен контрольный пример проектного решения, выполнена оценка экономической эффективности проекта.

В заключении представлены результаты выполненных в процессе работы задач.

Глава 1 Функциональное моделирование предметной области

1.1 Характеристика деятельности предприятия оптово-розничной торговли

Оптовая торговля – это процесс продажи товаров в большом объеме и, как правило, по сниженной цене. Оптовая торговля включает в себя покупку товаров на оптовых складах и/или напрямую у производителя, а также последующую продажу этих товаров розничным продавцам, торговым сетям, предприятиям и другим оптовым покупателям.

Розничная торговля – это процесс продажи товаров конечным потребителям для личного пользования в небольших объемах и, как правило, по более высоким ценам, чем в оптовой торговле. Розничная торговля направлена на удовлетворение потребностей конечных потребителей в широком ассортименте товаров и предоставляет им возможность совершать покупки в удобных для них местах (рынки, магазины, торговые развлекательные центры).

Предприятие оптово-розничной торговли – это коммерческая организация, которая занимается продажей товаров как оптом, так и в розницу. Оно может иметь свои собственные магазины, супермаркеты и интернет-магазины, а также, как правило, имеет свои собственные склады и логистические системы, которые позволяют ему эффективно управлять запасами товаров и осуществлять доставку товаров покупателям. Такое предприятие может работать в различных отраслях.

Организационная структура каждого отдельного предприятия оптово-розничной торговли может варьироваться в зависимости от размера этого предприятия, специфики ведения бизнеса, ассортимента товаров и многих других факторов. Однако, как правило, оно включает в себя перечисленные на рисунке 1 подразделения.

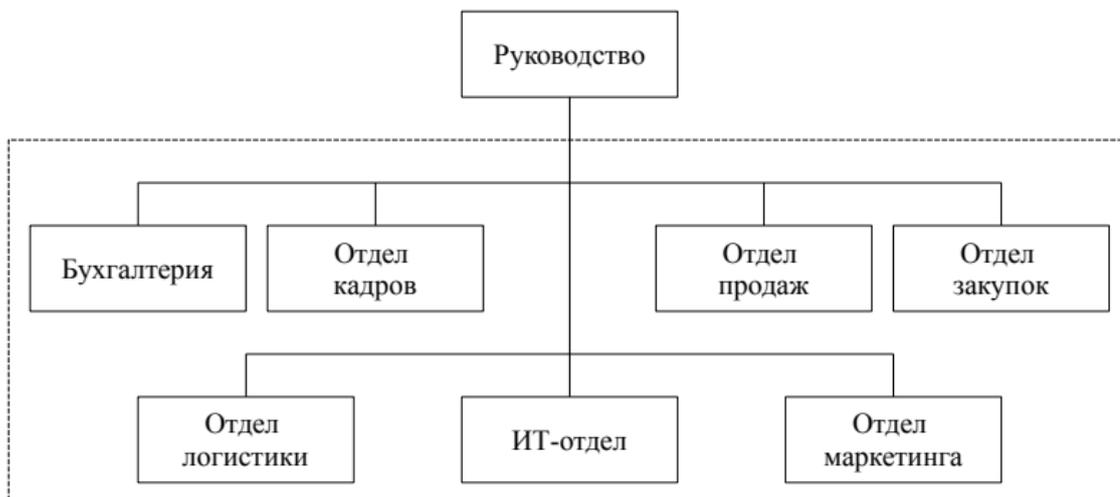


Рисунок 1 – Диаграмма организационной структуры предприятия оптово-розничной торговли

Каждое из подразделений предприятия выполняет важную роль в его правильном и эффективном функционировании.

Руководство выполняет следующие функции:

- формирование стратегии – определение направлений развития предприятия и принятие стратегических решений, анализ рисков и управление ими;
- формирование бюджета и контроль за его использованием;
- формирование благоприятного имиджа – организация взаимодействий с внешними партнерами, формирование бизнес-контактов.

Отдел управления персоналом выполняет следующие функции:

- подбор персонала – разработка требований к кандидатам на вакансии, поиск и отбор подходящих кандидатов, проведение собеседований;
- развитие и обучение персонала – разработка и внедрение программ обучения, организация тренингов и семинаров на предприятии;

- мотивация персонала – разработка систем оценки производительности, определение заработной платы и уровня поддержки персонала;
- управление персональными данными – ведение учета и хранение данных о персонале, анализ данных и составление отчетов для руководства;
- соблюдение законодательства в области труда и социального обеспечения.

Отдел бухгалтерии выполняет следующие функции:

- управление бюджетом предприятия – участие в формировании бюджетного плана, контроль за его соблюдением и выявление отклонений;
- осуществление платежей сотрудникам и контрагентам – подготовка и отправка платежных документов, а также контроль за исполнением платежей;
- ведение финансового учета – регистрация финансовых транзакций, их обработка и формирование отчетов для руководства и налоговой;
- соблюдение законодательства в области налогообложения.

Отдел закупок выполняет следующие функции:

- планирование закупок – анализ потребностей в товарах, разработка годовых планов и бюджетов на закупки, оптимизация бюджетных затрат;
- поиск и выбор поставщиков – исследование рынка, поиск новых поставщиков, проведение тендерных процедур, оценка качества товаров и выбор наилучших поставщиков;
- оформление договоров – подготовка договоров на поставку товаров, контроль за выполнением условий, решение спорных вопросов;

- управление запасами – контроль за уровнем запасов товаров и оптимизация процесса закупок, предотвращение дефицита и профицита товаров;
- контроль качества товаров – контроль за качеством товаров, обеспечение соблюдения требований качества со стороны поставщиков;
- управление рисками – оценка рисков, связанных с закупками, разработка мер по снижению рисков, обеспечение безопасности и надежности поставок.

Отдел продаж выполняет следующие функции:

- привлечение клиентов – реклама продукции, участие в выставках, введение акций, распродаж и других мероприятий для привлечения клиентов;
- консультирование клиентов – предоставление информации о товарах и подбор оптимальных решений для клиентов;
- обработка заказов клиентов – оформление заказов, сборка товаров и, при необходимости, подготовка товаров к отправке и доставке;
- управление базой клиентов – поддержание связи с клиентами, разработка специальных предложений для них и управление клиентской базой данных;
- разработка стратегии продаж – контроль за выполнением планов продаж, разработка стратегий и планов развития, отслеживание тенденций и изменений на рынке, анализ конкурентов и рыночных трендов.

Отдел маркетинга выполняет следующие функции:

- исследование рынка – анализ рынка, конкурентов, потребителей и трендов;

- разработка маркетинговой стратегии – определение целевой аудитории, позиционирования продукта на рынке и стратегии продвижения;
- реклама и продвижение – создание рекламных кампаний и материалов, создание образа бренда и управление репутацией компании;
- анализ эффективности продвижения – сбор данных о продажах и отзывах от клиентов, оценка эффективности маркетинговых стратегий;
- связь с клиентами – общение с клиентами, сбор отзывов и предложений, разработка программ лояльности и удержания клиентов;

Отдел логистики выполняет следующие функции:

- управление складом – контроль за уровнем запасов товаров на складе, организация работы склада, учет поступления и отгрузки товаров;
- планирование и организация поставок товаров – планирование маршрутов и способа доставки, транспортировка товаров на склад и покупателю;
- связь с поставщиками и покупателями – для организации доставки товаров, согласования условий доставки и оплаты.

ИТ-отдел выполняет следующие функции:

- поддержка информационной инфраструктуры предприятия;
- улучшение бизнес-процессов на предприятии;
- создание и/или внедрение различных АИС (автоматизированных информационных систем), которые уменьшают объем рутинных активностей и выполняют автоматизированные действия, увеличивая эффективность и производительность всех сотрудников на предприятии.

Ознакомившись с организационной структурой предприятия оптово-розничной торговли и функциями, выполняемыми его подразделениями, можно составить следующий список верхнеуровневых бизнес-процессов:

- процесс управления финансами: планирование, контроль и анализ;
- процесс управления персоналом: прием на работу, обучение, оценка и управление;
- процесс маркетинга: планирование, разработка и реализация стратегий;
- процесс логистики: организация и контроль перемещения товаров;
- процесс закупок: закупка товаров для продажи;
- процесс продаж: продажа товаров клиентам;
- процесс IT-управления: разработка и поддержка информационной инфраструктуры предприятия.

Наиболее важным бизнес-процессом в рамках этой выпускной квалификационной работы является бизнес-процесс продаж, который детально рассмотрим в следующих параграфах, а также выполним подробную декомпозицию этого процесса.

Перед тем как переходить к бизнес-процессу продаж – будет полезно разобрать бизнес-процесс закупок для лучшего понимания общей картины происходящего, так как процесс продаж следует за процессом закупок, а также процесс закупок не изменяется в зависимости от наличия или отсутствия веб-представительства на предприятии оптово-розничной торговли [2, 5, 9, 25].

Функциональная модель бизнес-процесса закупок представлена на рисунке 2.

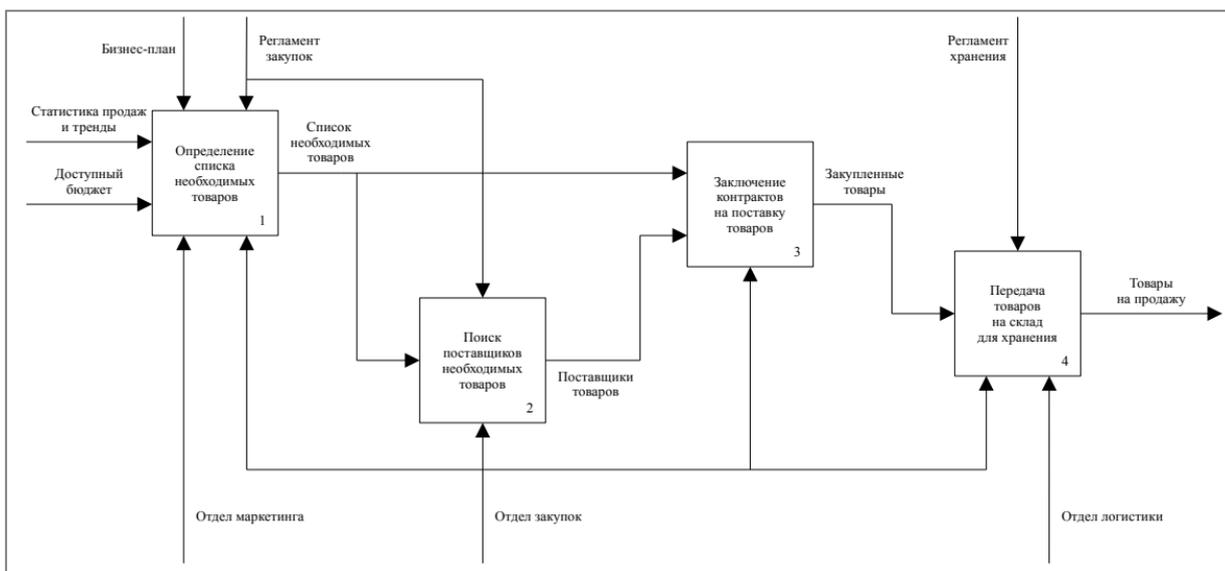


Рисунок 2 – Функциональная IDEF0-диаграмма бизнес-процесса закупки товаров на предприятии оптово-розничной торговли

Можно выделить четыре основных сценариев использования в бизнес-процессе закупок:

- определение списка необходимых товаров для закупки согласно статистике продаж, трендам и доступному бюджету;
- поиск поставщиков, которые могут предоставить необходимые для закупки товары;
- заключение контрактов на поставку необходимых товаров от поставщиков;
- получение заказанных товаров от поставщиков и передача их на склад для хранения и последующей транспортировки в магазины и/или покупателям.

Основными акторами (действующими лицами) в бизнес-процессе закупок товаров можно выделить: отдел закупок, отдел маркетинга, отдел логистики.

Отображенная и описанная выше диаграмма закупок позволит более качественно разобрать последующие бизнес-процессы, которые будут разработаны в следующих параграфах.

1.2 Разработка функциональной модели бизнес-процессов AS-IS и обоснование необходимости автоматизированного варианта решения

1.2.1 Разработка функциональной модели бизнес-процесса управления поставками товара

Перед тем как переходить непосредственно к описанию и разработке диаграмм для бизнес-процесса продаж разберем промежуточный бизнес-процесс управления поставками товара, находящийся между бизнес-процессом закупок и бизнес-процессом продаж.

В рамках бизнес-процесса управления поставками товара можно выделить следующих акторов: отдел логистики и отдел продаж.

Основными сценариями в рамках этого бизнес-процесса могут быть:

- передача товаров со склада в магазин;
- учет полученной партии товаров в системе учета (электронной или бумажной);
- выставление товара на витрину магазина.

Учитывая описанные выше пункты была разработана функциональная IDEF0-диаграмма бизнес-процесса управления поставками товара, изображенная на рисунке 3.

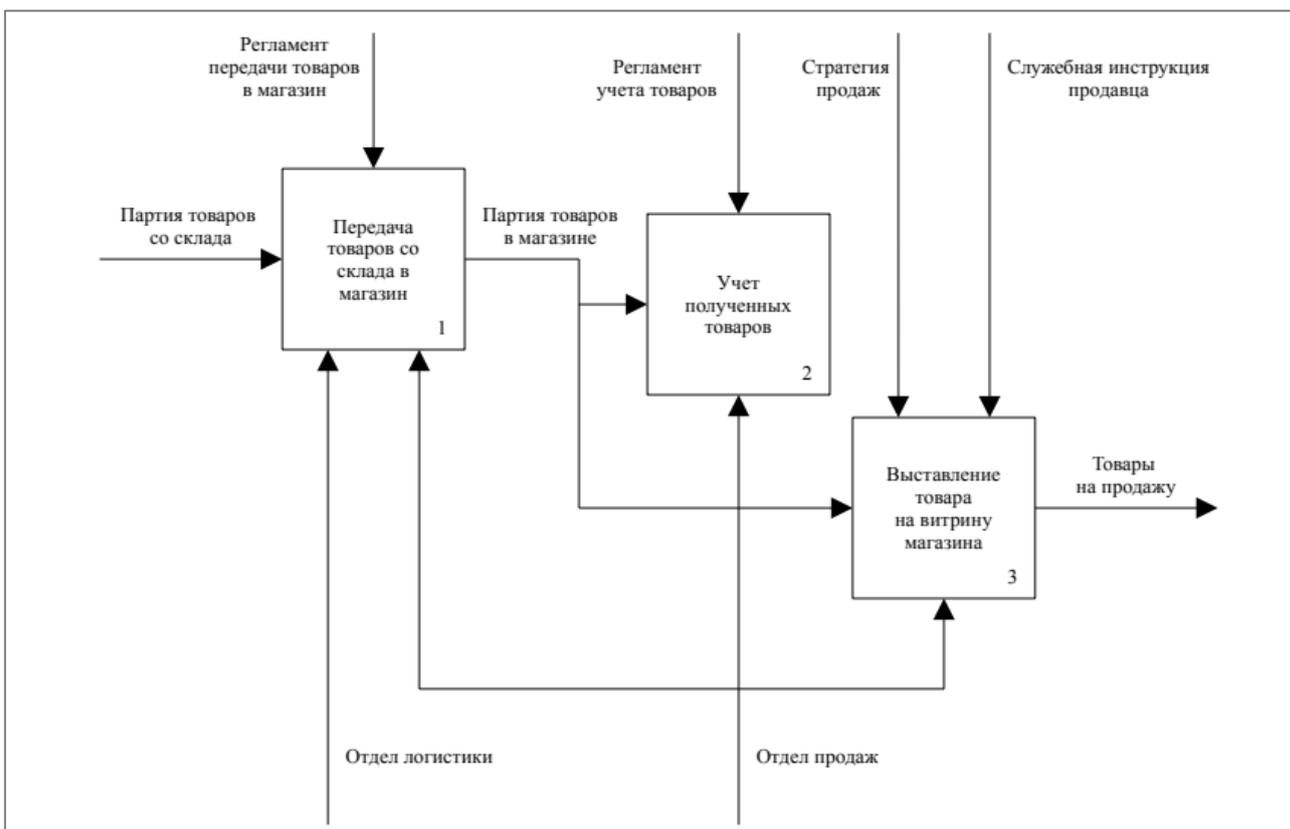


Рисунок 3 – Функциональная IDEF0-диаграмма бизнес-процесса управления поставками товара на предприятии оптово-розничной торговли без веб-представительства

Данный бизнес-процесс можно оптимизировать, сделав необязательными доставку товара в магазины и необходимость выставления этого товара на витрину. При интеграции веб-представительства эта оптимизация будет выполнена, поэтому далее по тексту этот бизнес-процесс и его оптимизация упоминаться больше не будут.

1.2.2 Разработка функциональной модели бизнес-процесса продаж

Основными акторами (действующими лицами) в бизнес-процессе продаж могут быть:

- покупатель (клиент – физическое или юридическое лицо);
- продавец (один из сотрудников предприятия – продавец-консультант, кассир, курьер).

Основными сценариями использования в бизнес-процессе продаж могут быть:

- ознакомление с каталогом товаров и информацией о товаре в магазине, а также консультация покупателя продавцом-консультантом;
- покупка желаемых товаров;
- получение приобретенных товаров на кассе или доставка товаров курьерской доставкой (при необходимости).

Согласно изложенным выше сценариям использования и действующим лицам была разработана функциональная IDEF0-диаграмма бизнес-процесса продаж на предприятии оптово-розничной торговли без веб-представительства, с которой можно ознакомиться на рисунке 4.

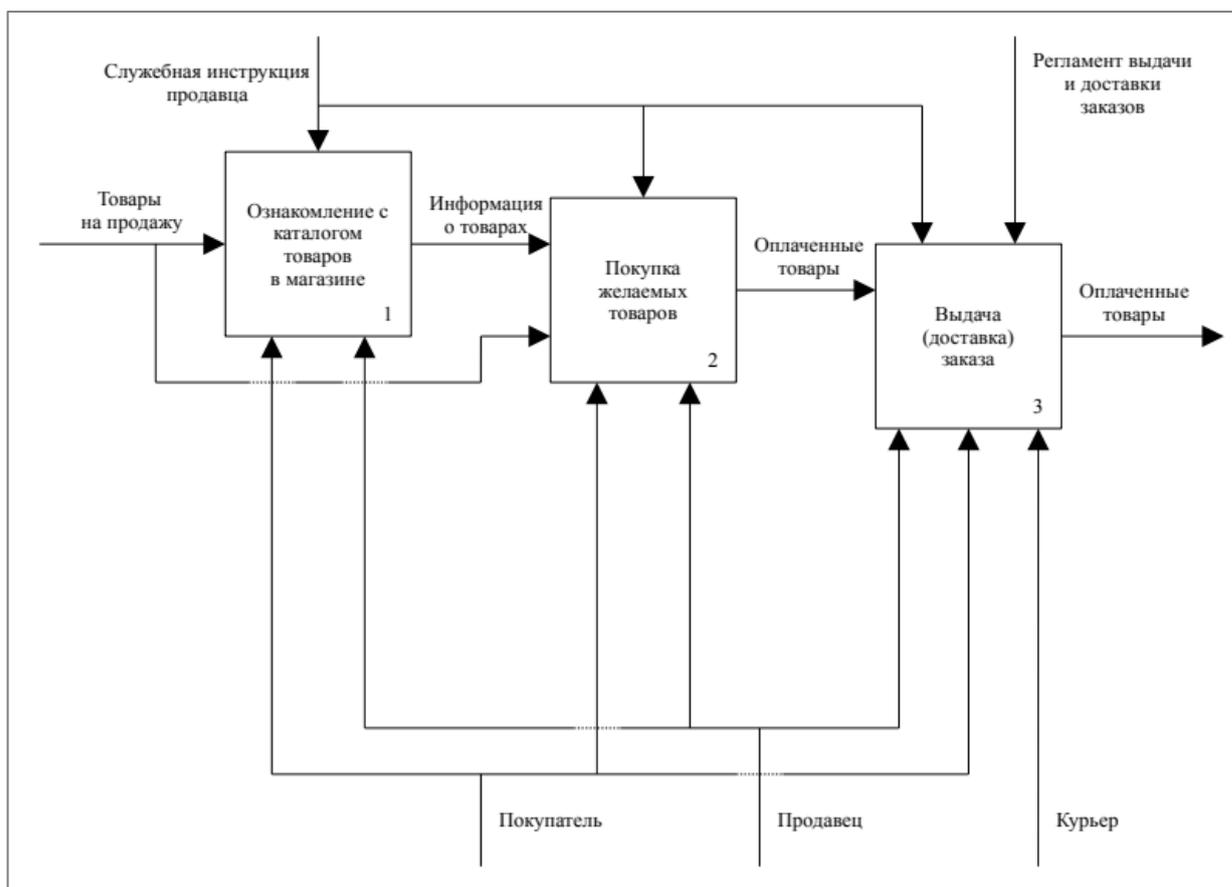


Рисунок 4 – Функциональная IDEF0-диаграмма бизнес-процесса покупки товаров покупателем на предприятии оптово-розничной торговли без веб-представительства

Данная диаграмма позволяет получить представление о бизнес-процессе продаж на предприятии и действующих лицах. Также, изучив диаграмму, можно выделить следующие недостатки текущего бизнес-процесса продаж:

- ограниченный охват аудитории – без веб-представительства предприятие ограничивает свой охват только территорией, на которой находится его магазин, и зависит от традиционных каналов продаж, таких как реклама в СМИ или персональные рекомендации (также знакомые как «сарафанное радио»), что значительно снижает потенциальные продажи;
- недостаточное количество информации – клиенты могут не иметь достаточно информации о товарах, которые предприятие предлагает, что может приводить к упущенным продажам;
- долгое время оформления заказов – оформление заказов по телефону очень неэффективное, так как требует от участников разговора относительно много коммуникаций и уточнений, прежде чем заказ будет оформлен. Оформление заказов в магазине же с другой стороны не страдает от подобной проблемы, но с другой стороны поход в магазин занимает время и может быть неудобным для клиентов, которые не могут посетить магазин в определенное время или в определенном месте;
- недостаточное взаимодействие с клиентами – без веб-представительства предприятие может упустить возможности для лучшего взаимодействия с клиентами, таких как: представление специальных персональных предложений, возможность предоставления клиентам дополнительной информации о продуктах, обратной связи и возможность онлайн-покупки, что может увеличить лояльность клиентов и увеличить продажи.

Из обозначенных выше проблем можно сделать вывод, что наличие веб-представительства для предприятия оптово-розничной торговли является необходимым условием для устранения недостатков в бизнес-процессе продаж, таких как ограниченный круг потенциальных клиентов, невозможность быстро и эффективно информировать клиентов о новых товарах, относительно большая стоимость рекламы в СМИ и т. д.

Автоматизация бизнес-процесса продаж позволит повысить эффективность работы предприятия, сократить затраты на рекламу и увеличить общую прибыльность предприятия.

1.3 Разработка функциональной модели бизнес-процессов ТО-ВЕ

Внедрение веб-представительства в значительной мере повлияет на бизнес-процесс продаж предприятия оптово-розничной торговли. Некоторые изменения, которые произойдут с бизнес-процессом продажи:

- процесс заказа товаров дополнится вариантом, когда клиенты смогут самостоятельно выбирать и заказывать товары через Интернет;
- появится необходимость обеспечения быстрой и качественной обработки заказов из интернет-магазина;
- возрастет нагрузка на отдел логистики, т. к. большую часть заказов потребуется доставлять покупателям службой курьерской доставки;
- потребуется продвижение товаров через интернет-маркетинг и рекламу;
- потребует дополнительное обучение сотрудников отдела продаж для работы с интернет-магазином и для обработки заказов, оформляемых через него;
- потребуется выполнить интеграцию интернет-магазина с внутренними системами управления предприятия для автоматизации процессов обработки заказов и учета продаж.

Учитывая вышеописанные изменения в бизнес-процессе продаж, на рисунке 5 составлена функциональная IDEF0-диаграмма бизнес-процесса продаж предприятия оптово-розничной торговли с веб-представительством.

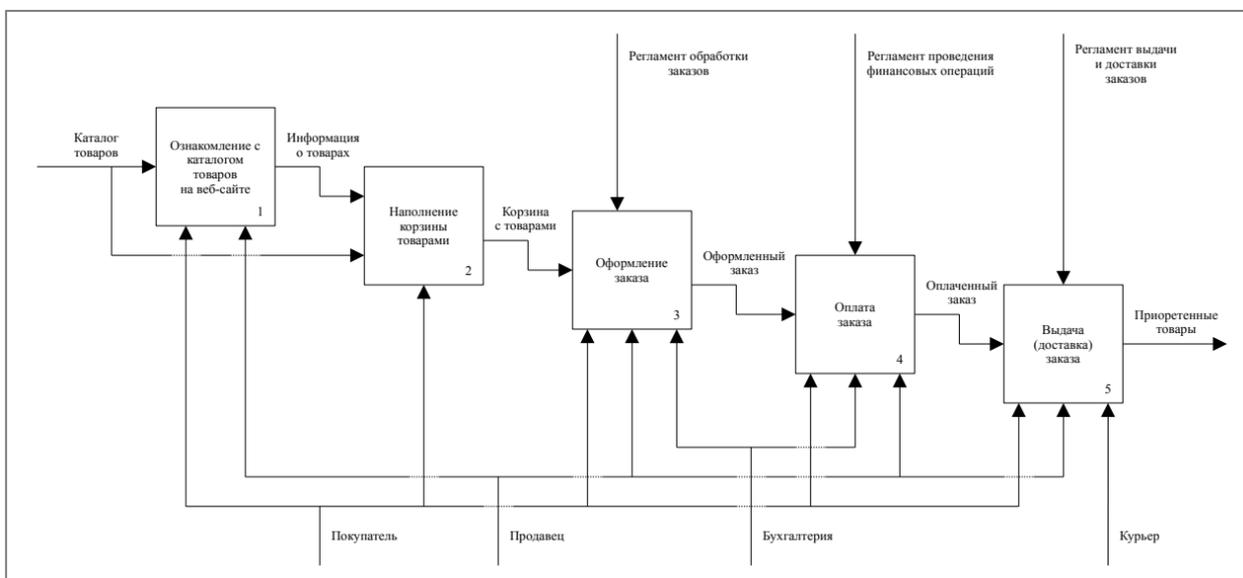


Рисунок 5 – Функциональная IDEF0-диаграмма бизнес-процесса покупки товаров покупателем на предприятии оптово-розничной торговли с веб-представительством

Бизнес-процесс продаж для предприятия с веб-представительством начинается с того, что потенциальный клиент находит сайт предприятия в Интернете и получает информацию о товарах, которые предлагает предприятие. Затем, если клиент заинтересован, он может оставить заявку на сайте или сделать заказ онлайн, предварительно добавив товары в корзину.

Отдел продаж получает заявки и заказы через систему управления продажами и связывается с клиентом для уточнения деталей заказа и его подтверждения. Если клиент нуждается в дополнительной информации, отдел продаж может предоставить её через электронную почту или телефон.

После того, как заказ был подтвержден, покупателю выставляется счет на оплату. После оплаты счета покупателем отдел продаж передает заказ на склад для сборки заказа. После сборки заказа он отправляется в отдел логистики на доставку к покупателю.

Весь процесс продаж отслеживается в системе управления продажами, что позволяет менеджерам получать данные о продажах и клиентах, анализировать эффективность процесса продаж и принимать управленческие решения для улучшения процесса.

1.4 Формирование требований к автоматизированному решению

Основываясь на выше изложенной информации, требования к автоматизированному решению для предприятия оптово-розничной торговли можно сформулировать следующим образом:

- веб-сайт должен быть легко доступным для потенциальных клиентов и должен иметь хороший пользовательский интерфейс и навигацию;
- веб-сайт должен содержать информацию о товарах, которые продает предприятие, а также информацию о ценах, скидках и акциях;
- веб-сайт должен иметь удобную систему онлайн-заказов с возможностью выбора товаров, оформления заказов, выбора способа оплаты и доставки;
- система управления контентом (CMS, Content Management System) должна быть легко доступной и удобной для использования, чтобы персонал мог быстро обновлять информацию о товарах, ценах и скидках;
- интеграция веб-сайта с системой управления заказами и инвентаризации должна быть гладкой, чтобы заказы могли обрабатываться автоматически и быстро;
- веб-сайт должен быть безопасным и защищенным от взломов и кражи персональных данных клиентов;

- сайт должен быть адаптивным и оптимизированным для мобильных устройств, чтобы клиенты могли легко совершать заказы с любого устройства;
- необходимо включать функциональность отзывов и рейтингов, чтобы клиенты могли оставлять отзывы о товарах;
- веб-сайт должен быть оптимизирован для поисковых систем, чтобы повысить его видимость и привлечь больше клиентов;
- необходимо предоставить инструменты для анализа поведения клиентов на сайте и их покупательных привычек, чтобы улучшить пользовательский опыт и повысить эффективность продаж.

Основываясь на этих требованиях можно реализовать качественное и полнофункциональное веб-представительство.

1.5 Анализ существующих разработок на предмет соответствия сформулированным требованиям

Для реализации веб-представительства можно использовать различные программные продукты и решения, такие как:

- CMS (Content Management System) – системы управления контентом, такие как WordPress, Drupal, Joomla. Они позволяют легко создавать и управлять веб-сайтом, добавлять и редактировать контент, а также имеют широкие возможности для расширения функциональности;
- E-commerce (Electronic Commerce) платформы – такие как Magento, OpenCart, PrestaShop. Они специализированы на создании интернет-магазинов и имеют встроенные функции для управления каталогом товаров, заказами и оплатой;
- CRM (Customer Relationship Management) системы – такие как Bitrix24, Salesforce, Zoho CRM. Они позволяют автоматизировать

- процессы продаж и управления клиентами, собирать и анализировать данные о клиентах, управлять заказами и платежами;
- системы электронной коммерции, такие как Shopify, BigCommerce, Wix. Они предоставляют функциональность для создания интернет-магазинов, платежных систем, систем управления заказами и инвентаризации;
 - самописные веб-приложения на основе фреймворков и инструментов разработки, таких как Laravel, Ruby on Rails, Django.

Каждое из этих решений имеет свои преимущества и недостатки.

Выводы по главе 1

В результате функционального моделирования бизнес-процессов предприятия оптово-розничной торговли были выявлены проблемы и варианты решения этих проблем, которые могут быть реализованы с помощью внедрения веб-представительства и автоматизации бизнес-процессов. Было произведено аналитическое сравнение двух моделей бизнес-процессов: AS-IS и TO-BE. В результате были определены основные отличия между существующими процессами и тем, как они должны работать в идеальном случае.

Также были сформулированы требования к автоматизированному решению для интеграции веб-представительства. Проведенный анализ существующих разработок показал, что для реализации веб-представительства можно использовать различные решения, такие как платформы электронной коммерции, CRM-системы и т.д.

Таким образом, внедрение веб-представительства и автоматизация бизнес-процессов позволят предприятию оптово-розничной торговли улучшить качество обслуживания клиентов, оптимизировать логистические процессы и повысить эффективность работы всего предприятия.

Глава 2 Логическое проектирование

2.1 Логическая модель информационной системы и её описание

Логическая модель представляет собой высокоуровневое описание функций и компонентов системы, включая описание их взаимодействия. Она является промежуточным звеном между функциональной моделью бизнес-процессов и физической моделью информационной системы.

2.1.1 Постановка задачи логического моделирования

Определим цели и задачи, которые должна выполнять разрабатываемая информационная система [4, 11, 18, 19]:

- предоставить покупателям возможность ознакомиться с каталогом товаров через веб-сайт, а также оформить заказ через него;
- предоставить отделу продаж возможность управлять каталогом товаров и обрабатывать поступающие с веб-сайта заказы;
- предоставить сотрудникам предприятия возможность следить за статистикой продаж для корректировки стратегии закупок, продаж и маркетинга;
- предоставить бухгалтерии возможность формировать отчеты, сводки и прочие документы;
- предоставить отделу логистики возможность управлять заказами во время доставки товаров, а также контролировать его для улучшения процессов.

Учитывая поставленные выше цели можно перейти к описательной части компонентов.

2.1.2 Компоненты системы и их описание

Информационная система представляет совокупность компонентов, взаимодействующих между собой для обеспечения её функционирования:

- аппаратное обеспечение – это компонент, который представляет собой физические устройства, необходимые для функционирования ИС, такие как компьютеры, серверы, периферийные устройства, сетевое оборудование и прочие устройства;
- программное обеспечение (ПО) – это набор программных средств, необходимых для выполнения функций ИС. Он может включать в себя операционные системы, базы данных, программное обеспечение для защиты информации, сетевые сервисы и другие приложения.

Данные компоненты в свою очередь делятся на более мелкие подкомпоненты, которые играют важную роль при построении ИС.

2.1.3 Аппаратные компоненты системы

Наиболее важными компонентами аппаратного обеспечения являются сервер и клиент [17].

Сервер представляет собой высокопроизводительный компьютер, способный эффективно выполнять функциональную задачу ИС. Как правило, ИС строится не из одного сервера, а из множества серверов, которые имеют различную конфигурацию и выполняемые функции. Можно выделить следующие, наиболее важные функции, выполняемые сервером:

- хранение и обработка данных: сервер может служить как центральное хранилище данных, так и производить их обработку, включая выполнение вычислительных задач и запросов;
- управление сетью: сервер может выполнять функции маршрутизации, управления трафиком и защиты сети;
- управление пользователями и ресурсами: сервер может обеспечивать аутентификацию и авторизацию пользователей, а также управлять доступом к ресурсам системы;

- обеспечение надежности и безопасности: сервер может выполнять функции резервирования данных, контроля доступа, обнаружения и предотвращения атак и других угроз.

Сервер взаимодействует с другими компонентами ИС следующим образом:

- сервер может обращаться к базам данных и другим ресурсам, которые хранятся на других компонентах системы;
- клиенты, которые являются другими компонентами системы, могут обращаться к серверу для получения данных, выполнения вычислительных задач и других операций;
- сервер может взаимодействовать с другими серверами, если в системе используется распределенная архитектура;
- сервер может управлять другими компонентами системы, например, отправлять запросы на выполнение операций или управлять доступом к ресурсам.

Клиент же в свою очередь включает в себя все устройства, которые используются для доступа к ИС предприятия оптово-розничной торговли. Обычно это персональные компьютеры, но также могут быть планшеты, смартфоны и другие устройства, имеющие доступ к Интернету.

Основные функции клиента в информационной системе включают:

- клиент получает доступ к ИС для просмотра информации, которая хранится на сервере. Клиент может использоваться для просмотра каталога товаров, поиска информации о заказах, контактах с покупателями и других операций;
- клиент может использоваться для ввода данных в ИС, например, для оформления заказа или изменения информации в базе данных. Кроме того, клиент может отправлять запросы на сервер для обработки;

- клиент может использоваться для хранения информации, например, для сохранения данных заказов или истории покупок.

Клиент взаимодействует с другими компонентами ИС следующим образом:

- клиент запрашивает информацию: переходит на веб-сайт, чем запускает процесс запроса информации с сервера, после чего отображает её в интернет-браузере;
- клиент отправляет информацию: взаимодействует с элементами пользовательского интерфейса веб-сайта, тем самым выполняя действия и отправляя события и запросы на сервер;
- клиент обновляет информацию: после взаимодействий с веб-сайтом, тот в свою очередь обновляет веб-страницу, отображая актуальное состояние пользовательского интерфейса клиенту.

Таким образом, клиент и сервер взаимодействуют, чтобы обеспечить эффективное функционирование информационной системы предприятия оптово-розничной торговли.

2.1.4 Программные компоненты системы

Наиболее важными верхнеуровневыми программными компонентами являются наборы клиентского и серверного ПО.

Серверное ПО является одним из основных компонентов ИС на предприятии и выполняет ряд важных функций. Основными задачами такого ПО являются обеспечение хранения и обработки данных, а также управление доступом к этим данным. Оно отвечает за функционирование баз данных, веб-серверов, почтовых серверов, серверов приложений и других программных средств, необходимых для работы информационной системы.

Взаимодействие серверного ПО с другими компонентами ИС осуществляется через сетевые протоколы. Например, веб-серверное ПО может работать с клиентскими приложениями через протокол HTTP, а базы данных могут общаться с серверами приложений по протоколу JDBC.

Кроме того, серверное ПО может обеспечивать дополнительную функциональность, например, автоматическое создание резервных копий данных, мониторинг состояния системы, обнаружение ошибок. Оно также может предоставлять интерфейсы программирования приложений (API) для взаимодействия с другими приложениями и системами.

Клиентское ПО – это набор программных компонентов, которые устанавливаются на устройства пользователей и позволяют им взаимодействовать с сервером ИС. Оно выполняет следующие функции:

- предоставление интерфейса для работы с ИС: клиентское ПО обеспечивает пользователям доступ к функциям системы через графический интерфейс пользователя (GUI). Например, в оптовой розничной торговле, клиентское ПО может позволять пользователям оформлять заказы, просматривать каталоги товаров, просматривать информацию о заказах;
- хранение и обработка данных на стороне клиента: клиентское ПО может также выполнять некоторые операции над данными на стороне клиента. Например, оно может сохранять данные о заказах, истории покупок и т.д. на устройстве пользователя, чтобы обеспечить быстрый доступ к этой информации;
- взаимодействие с сервером ИС: клиентское ПО общается с сервером, оно может отправлять запросы на сервер для получения данных, обрабатывать ответы от сервера и взаимодействовать с другими компонентами ИС через сетевые протоколы. Для обмена данными между клиентским и серверным ПО могут использоваться различные форматы передачи данных, такие как XML, JSON и другие;
- обеспечение безопасности – клиентское ПО может выполнять функции аутентификации пользователей, защиты передаваемых данных и контроля доступа к ресурсам системы. Например, в

оптово-розничной торговле клиентское ПО может требовать от пользователя ввода логина и пароля, чтобы получить доступ к личной информации и истории заказов.

2.2 Разработка логической модели данных

Логическая модель данных описывает структуру данных в системе и связи между ними. Эта модель является абстрактной и не зависит от конкретных технологий и СУБД, что позволяет легко переносить систему на другие платформы [1].

2.2.1 Сущности базы данных

В рамках этого пункта выделены сущности БД, определены их свойства и связи. На рисунке 6 представлена UML-диаграмма классов [3, 12, 21].

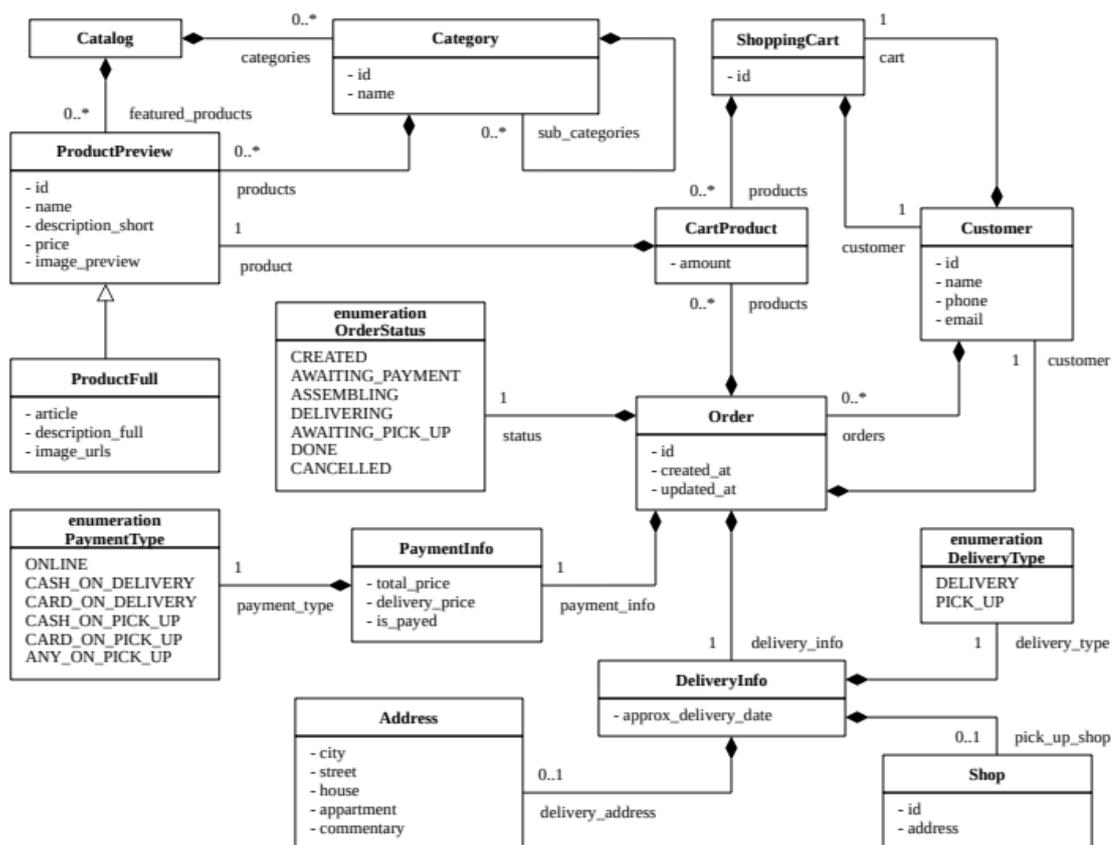


Рисунок 6 – UML-диаграмма сущностей АИС

Рассмотрим более детально представленные на диаграмме сущности.

Сущность «Товар» имеет следующие свойства:

- уникальный идентификатор товара в ИС;
- наименование товара;
- артикул товара;
- описание товара;
- цена товара;
- список изображений товара;
- количество товара на складе.

Сущность «Товар» используется в нескольких сущностях:

- в каталоге товаров в качестве продвигаемых товаров;
- в категориях товаров;
- в корзине, при этом каждый товар имеет связку с целочисленным значением, обозначающим количество товара, добавленного в корзину;
- в созданных заказах покупателей.

Стоит уточнить, что сущность «Товар» имеет базовую форму в виде «ProductPreview» и расширенную форму в виде наследника «ProductFull» – такое разделение необходимо для оптимизации объема данных, передаваемых по сети Интернет в запросах к серверу и ответах от сервера, а также для упрощенного алгоритма сбора информации для каталога товаров.

Сущность «Каталог товаров» состоит непосредственно из самих товаров, которые поделены на категории, а также, как правило, каталог содержит определенный набор продвигаемых товаров, которые предприятие намеревается выделить отдельно от других товаров в рамках маркетинговой стратегии.

Сущность «Категория товаров» в свою очередь имеет: идентификатор, наименование, список товаров и другие вложенные категории товаров,

которые позволяют строить древовидную структуру каталога товаров, в которой пользователь может быстро находить необходимые ему товары, подходящие по контексту категории.

Сущность «Покупатель» используется в случае, если он авторизован, и имеет следующие свойства:

- уникальный идентификатор покупателя в ИС;
- наименование покупателя;
- номер телефона;
- адрес электронной почты;
- идентификатор корзины с товарами;
- список заказов.

В случае, если пользователь не авторизован в ИС, то используется локальное хранилище товаров корзины на клиентском устройстве, которые будут переданы в пользовательскую корзину после авторизации или при оформлении заказа.

Сущность «Заказ» содержит следующие свойства:

- уникальный идентификатор заказа;
- статус заказа;
- временные штампы «когда заказ создан» и «когда заказ обновлен»;
- информация о доставке;
- информация об оплате.

Описанные выше сущности составляют основу логической модели ИС.

2.2.2. Моделирование процесса наполнения корзины товаров

Для процесса наполнения корзины товаров разработана диаграмма последовательности, отображенная на рисунке 7.



Рисунок 7 – Диаграмма последовательности наполнения корзины товаров

Также дополним диаграмму блок-схемой, на которой будет аналогично отображен алгоритм наполнения корзины товаров, результат на рисунке 8.

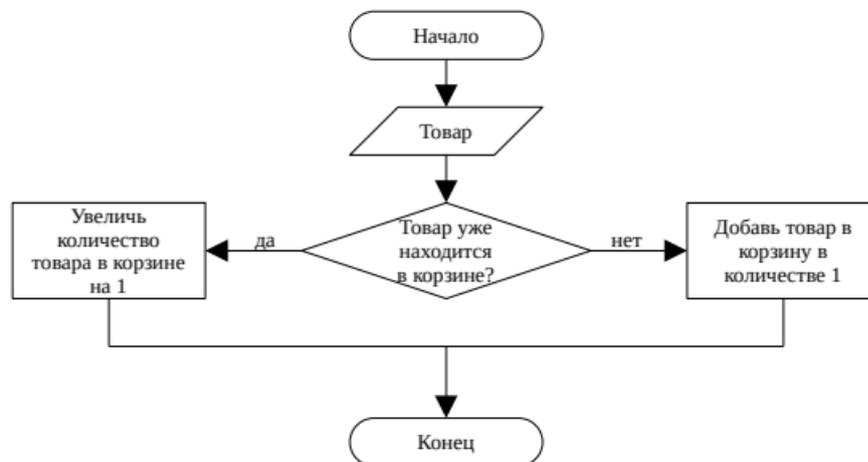


Рисунок 8 – Блок-схема алгоритма наполнения корзины товаров

Аналогичным способом, но в обратную сторону, будет работать алгоритм удаления товаров из корзины.

2.3 Информационное и техническое обеспечение

Интерфейс веб-сайта можно представить схематично, в виде рисунка, состоящего из абстрактных блоков, которые определяют примерный внешний вид разрабатываемой системы – такой интерфейс, как правило, называют каркасом. Каркасы интерфейса могут помочь более детально описать используемые в рамках системы модели и более качественно вникнуть в предметную область, а также найти потенциальные проблемные места на более раннем этапе разработки [14, 15].

Разработаем поверхностный каркас веб-сайта. Он должен содержать наиболее верхнеуровневые элементы (в плане вложенности), которые видны пользователю все время взаимодействия с веб-сайтом [23]:

- наименование веб-сайта (как правило, сопровождается логотипом);
- строка поиска по сайту – для быстрого поиска интересующих покупателя товаров;
- главное меню с горизонтальным блоком кнопок – для предоставления возможности осуществления быстрой навигации к наиболее важным страницам веб-сайта;
- сводка о пользователе и корзина – блок, который предлагает пользователю авторизоваться, если он еще не авторизован, в ином случае отображает информацию об авторизованном пользователе и позволяет перейти на веб-страницы с информацией об авторизованном пользователе, а также содержит краткую сводку по корзине и кнопку для перехода к странице с корзиной, в которой находятся товары, добавленные в неё ранее покупателем;
- боковое меню с вертикальным блоком кнопок – содержит древовидный список категорий товаров, определяющий полный каталог товаров, доступный покупателю к приобретению;

- тело сайта – отображает выбранную в данный момент страницу, в которой динамически изменяется наполнение.

Согласно вышеописанным требованиям разработан каркас веб-сайта, который можно изучить на рисунке 9.

Наименование сайта	Строка поиска по сайту	Сводка о пользователе и корзина
Главное меню, содержащее горизонтальный список кнопок для быстрого перехода к популярным и важным разделам сайта		
Боковое меню, содержащее вертикальный древовидный список категорий товаров	Тело сайта в котором отображается список товаров или любая другая информация, в зависимости от текущего местонахождения пользователя на сайте	

Рисунок 9 – Поверхностный каркас веб-сайта

Далее выполним более детальную разработку каркаса для тела сайта, который даст больше информации о том, какой контент будет находиться на веб-сайте.

Начнем с разработки тела веб-сайта, в котором находится каталог товаров, доступных для приобретения. Такой каталог, как правило, представляют в виде списка или сетки, в которых каждая ячейка определяет один товар. Для каждого товара в каталоге будет свойственно иметь:

- изображение товара;
- наименование товара;

- краткое описание товара;
- цену товара в количестве 1 шт;
- кнопку «добавить в корзину».

Учитывая определенные выше свойства разработан каркас каталога товаров в виде сетки, с которым можно ознакомиться на рисунке 10.

Изображение		Изображение		Изображение	
Наименование		Наименование		Наименование	
Описание		Описание		Описание	
Цена	В корзину	Цена	В корзину	Цена	В корзину
Изображение		Изображение		Изображение	
Наименование		Наименование		Наименование	
Описание		Описание		Описание	
Цена	В корзину	Цена	В корзину	Цена	В корзину

Рисунок 10 – Каркас тела веб-сайта, в котором находится каталог товаров

У каждого элемента (товара) в каталоге есть интерактивные элементы. Как правило, при нажатии на ячейку товара – открывается отдельная страничка на веб-сайте, на которой описана детальная информация о товаре, а также при нажатии на кнопку «добавить в корзину» – выполняется действие добавления товара в корзину (в локальную корзину устройства, если пользователь не авторизован, либо в удаленную корзину на сервере, если пользователь авторизован).

Также, опционально, можно отображать у товаров в каталоге количество этого товара в корзине в виде счетчика, чтобы пользователь мог определить, какие товары находятся в его корзине без необходимости выполнять дополнительные шаги в виде перехода в корзину и обратно к каталогу товаров.

Далее опишем каркас тела веб-сайта, в котором находится детальное описание товара с более подробной информацией, чем она изложена в каталоге.

Детали товара включают в себя:

- наименование товара;
- подробное описание товара, включающее в себя полный перечень характеристик и параметров, важных для покупателя;
- несколько изображений в высоком качестве, между которыми можно переключаться с помощью стрелочек и миниатюр;
- цена товара;
- кнопка «добавить в корзину».

Нажатие на изображение товара открывает его на весь экран, нажатие на миниатюру или стрелочки в блоке изображения переключает активное изображение товара. Нажатие на кнопку «добавить в корзину» добавляет товар в корзину, также аналогично каталогу товара здесь может быть счетчик для отображения количества товара, находящегося в корзине.

Согласно описанию разработан каркас тела веб-сайта, в котором находится детальное описание товара, с которым можно ознакомиться на рисунке 11.

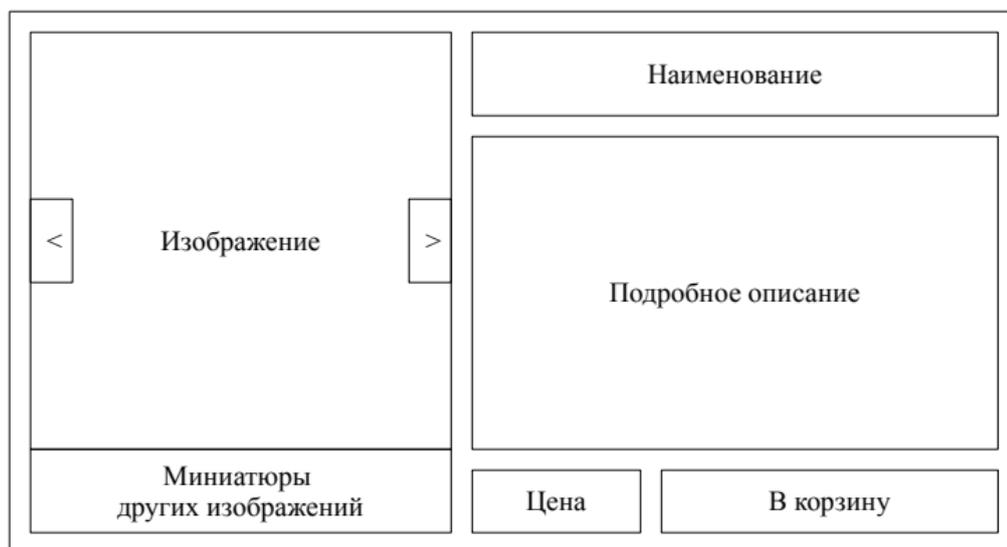


Рисунок 11 – Каркас тела веб-сайта, в котором находится детальное описание товара

После детального описания товара будет полезным разработать корзину товаров, в которой пользователь может ознакомиться с составом заказа, а также оформить его. Корзина товара включает в себя:

- список товаров, добавленных ранее в корзину покупателем. Покупатель может управлять количеством товара с помощью кнопок «-» и «+», убавляя или прибавляя количество товара, а также может удалить товар из корзины с помощью кнопки «X»;
- блок с информацией по заказу, в которой можно управлять параметрами доставки и оплаты, а также отображено количество товара в корзине, цена доставки и итоговая цена заказа;
- кнопка «оформить заказ», которая отправляет запрос на сервер и при необходимости перенаправляет на платежный шлюз, если выбрана онлайн оплата, либо перенаправляет на страничку с информацией по оформленному заказу.

Согласно этому описанию разработан каркас корзины товаров, с которым можно ознакомиться на рисунке 12.

Изображение товара №1	Наименование товара №1	X	
	Описание товара №1	Цена товара №1	
	-	Количество	+
Изображение товара №2	Наименование товара №2	X	
	Описание товара №2	Цена товара №2	
	-	Количество	+
Заголовок блока: Информация по заказу			
Контактный номер телефона и адрес электронной почты (если покупатель не авторизован) Тип доставки (самовывоз или курьер) Адрес доставки (магазина или местоположения) Тип оплаты (онлайн или при получении) Цена доставки и итоговая цена заказа Прочая информация о заказе			
Оформить заказ			

Рисунок 12 – Каркас тела веб-сайта, в котором находится корзина товаров

Разработанных каркасов интерфейсов веб-сайта должно хватить для лучшего понимания предметной области и описанных логических моделей.

Выводы по главе 2

В результате выполненной работы по описанию и разработке логических моделей ИС получен достаточный объем информации, необходимый для начала непосредственной разработки ИС.

Описана логическая модель и компоненты системы, разработаны абстрактные каркасы интерфейса веб-сайта, сущности БД и взаимосвязи между ними согласно логической модели системы.

Глава 3 Физическое проектирование

3.1 Системная архитектура проекта

Для реализации веб-представительства для предприятия оптово-розничной торговли была выбрана трехзвенная архитектура клиент-сервер. Основными преимуществами использования такой архитектуры являются:

- распределение функций между слоями: трехзвенная архитектура позволяет распределить функции между слоями ПО и упростить их разработку, поддержку и модификацию;
- простота внесения изменений: трехзвенная архитектура позволяет лучше поддерживать систему в целом, так как каждый слой может быть модифицирован или заменен без влияния на другие слои;
- масштабируемость: каждый слой может быть масштабирован независимо от других слоев. Это позволяет лучше управлять нагрузкой на систему и обеспечивать высокую производительность;
- улучшенная безопасность: каждый слой может быть защищен своей собственной системой безопасности, что повышает безопасность всей ИС.

Для выбранной трехзвенной архитектуры разработана UML-диаграмма пакетов АИС, с которой можно ознакомиться на рисунке 13.

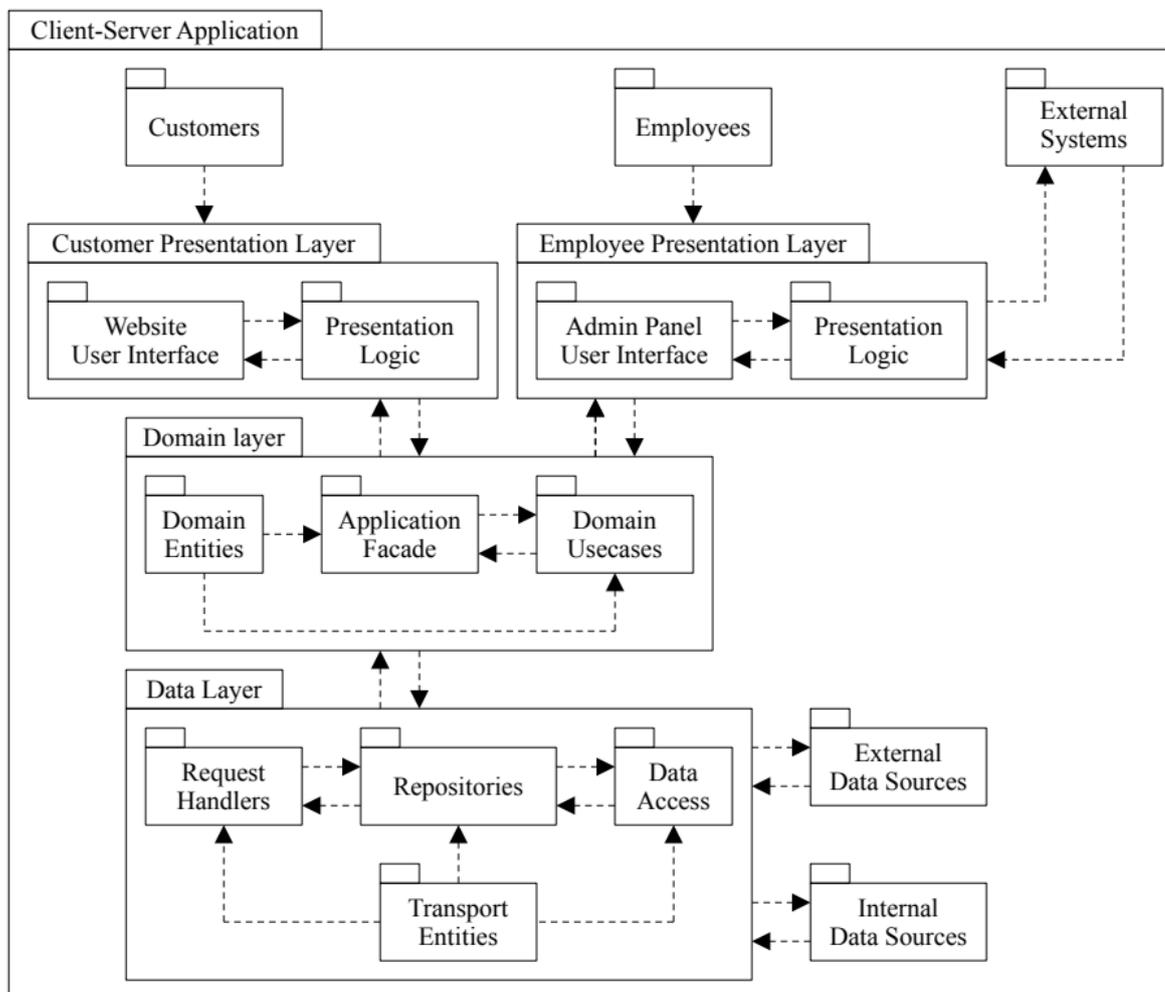


Рисунок 13 – UML-диаграмма пакетов АИС

Входными точками в АИС являются компоненты: «Customers» или «Employees» – это пользователи системы, которые входят в АИС посредством веб-браузера или другого предусмотренного для взаимодействия с системой ПО. Каждый из пользователей, в зависимости от точки входа, попадает в свой выделенный компонент: «Customer Presentation Layer» для покупателя и «Employee Presentation Layer» для работника предприятия.

«Customer Presentation Layer» отвечает за отображение информации пользователю на веб-сайте, а также за обработку пользовательского ввода для реализации возможности оформить заказ.

«Employee Presentation Layer» в свою же очередь отвечает за отображение служебной информации по наполнению веб-сайта (каталога товаров и информации о самих товарах), о заказах, о клиентах и прочей связанной с заказами информацией, а также обрабатывает пользовательский ввод для управления вышеперечисленными элементами системы.

«Domain Layer» реализуется в рамках веб-сервера, с которым общается «Presentation Layer» покупателя и сотрудника предприятия, и отвечает за формирование веб-страницы, которую видят пользователи при посещении веб-сайта, а также за обработку запросов от веб-сайта и формирование ответов для реакции на действия пользователей и других компонентов АИС.

«Domain Layer» содержит в себе:

- доменные модели данных («Domain Entities»), которые описывают прикладной уровень приложения в виде набора классов и связанных с ними свойств, которые используются для этого уровня;
- фасад («Application Facade»), который обрабатывает запросы от пользователя и делегирует их обработку определенным обработчикам («Domain UseCases»);
- обработчик/сценарии из компонента «Domain UseCases» является делегатом, который позволяет для каждого действия пользователя выделять относительно независимый компонент бизнес-логики, реализующий определенное для этого функционала поведение.

«Domain Layer» при этом напрямую не знает о базах данных, их таблицах и других подробностях хранения и передачи данных, он зависит лишь от интерфейсов сущностей, которые предоставляют уже трансформированные доменные данные наружу из более низкого, транспортного уровня «Data Layer».

«Data Layer» выполняет функции по реализации низкоуровневых CRUD-операций (Create, Read, Update, Delete), передачи данных между компонентами системы, а также, как правило, по трансформации

транспортных моделей данных в доменные модели данных, которые могут быть использованы в «Domain Layer» без предварительной обработки.

«Data Layer» содержит в себе:

- «Transport Entities» компонент составляет набор сериализуемых моделей данных, использующихся для передачи информации по различным линиям передачи данных (локальная сеть, сеть Интернет, межпроцессорное взаимодействие);
- «Request Handler» компонент, который принимает входящие запросы, выполняет предварительную обработку данных и передает ее в «Domain Layer» компоненту «Application Facade» для последующей обработки запроса;
- «Repositories» компонент выступает в роли абстракции над методом получения данных (или источника данных) и выполняет операцию по трансформации транспортных моделей данных в доменные модели данных. Репозитории из этого компонента, как правило, используются в «Domain Layer» в компоненте «Domain UseCases», и являются единственным источником данных для юзкейсов;
- «Data Access» компонент выступает в роли дополнительной абстракции над конкретным методом получения данных, например можно выделить «RemoteCustomerDataSource» и «LocalCustomerDataSource», которые будут соответственно брать информацию либо из удаленного источника данных посредством выполнения запроса к другому компоненту АИС, либо из закешированных данных.

3.2 Выбор СУБД и программных средств веб-разработки

Для веб-серверов можно использовать различные операционные системы, в зависимости от требований, и предпочтений системных

администраторов и разработчиков серверного ПО, но фактически отраслевым стандартом являются операционные системы на базе ядра Linux, такие как Debian, Ubuntu, RedHat и другие. Наиболее стабильным, надежным и проверенным временем дистрибутивом является Debian, который будем использовать для хостинга веб-сервера.

Для разработки реального (production) веб-сервера могут использоваться различные языки программирования и фреймворки, удобные команде разработки, но поскольку разработка веб-сервера не является целью данной ВКР, то ниже будет описан стек для небольшого тестового веб-сервера, используемого в рамках этой ВКР для тестирования функционала веб-сайта.

В первую очередь такой сервер выступает в роли «бекенда для фронтенда» (Back-end for Front-end) – это промежуточное звено между реальными поставщиками данных (другими серверами и БД) и веб-клиентом. Он, как правило, используется в случаях, когда необходим легковесный, так называемый тонкий клиент (Thin client). Веб-сервер выполняет всю тяжелую работу по сбору и препроцессингу данных, после чего передает их на клиент, клиенту же в свою очередь необходимо только выполнять запросы к серверу и обрабатывать его ответы в соответствии с разработанной спецификацией.

В качестве языка программирования для реализации тестового веб-сервера выбран Kotlin. Он является полностью совместимым с Java и работает на виртуальной машине Java (JVM), соответственно это позволяет переиспользовать огромное количество библиотек и кода в целом, написанного на Java. Кроме того Kotlin является очень мощным и современным языком программирования с удобными инструментами разработки, что позволяет достигать большой скорости разработки при этом соблюдая высокое качество разрабатываемого программного обеспечения.

Разработка на языке Kotlin, как правило, ведется в интегрированной среде разработки IntelliJ IDEA, поскольку эта среда предоставляет

наилучшую интеграцию с Kotlin и семейством языков, работающих на JVM в целом [8, 10].

Система сборки будет использоваться Gradle, как наиболее популярная и расширяемая система сборки для языков, работающих на JVM. Она позволит разделить код на переиспользуемые базовые модули, а также на независимые модули-сервисы при необходимости, что позволит изолировать код от нежелательных изменений и упростить командную разработку проекта.

В качестве веб-сервера будет использоваться Ktor, который является легковесным, простым в использовании и при этом расширяемым веб-сервером, предоставляющим прочный фундамент, а также огромное количество возможностей для расширения посредством установки различных плагинов [7].

В качестве СУБД будет использоваться MySQL, как одна из самых распространенных и простых в использовании, при этом имеющую хорошую скорость работы и масштабируемость.

Для контроля версий исходного кода будет использоваться Git, который также является отраслевым стандартом в сфере разработки ПО.

Сервис, который будет выполнять роль хостинга репозитория с исходным кодом компонентов веб-сервера, можно выбирать исходя из личных предпочтений команды разработки, т. к. git-репозитории можно с легкостью перемещать и/или зеркалировать между несколькими хостингами. Хорошими вариантами будут GitHub, GitLab, Bitbucket или выделенный сервер предприятия, на котором запущен собственный git-сервер.

Непосредственно сам веб-сайт будет реализован с использованием паттерна SPA (Single Page Application) – это паттерн реализации веб-приложения, который загружает одну единственную HTML-страницу с сервера, после чего динамически обновляет эту страницу (это также

относится и к навигации на другие разделы веб-сайта – наполнение обновляется динамически). Ниже перечислены плюсы такого выбора:

- отзывчивость и лучший пользовательский опыт: нет нужды постоянно перезагружать страницу веб-сайта в ответ на действия пользователя, т. к. информация обновляется динамически, что предоставляет пользователям привычный опыт, схожий с использованием другого нативного ПО;
- простота разработки: SPA позволяет упростить работу разработчикам, поскольку фреймворки SPA, как правило, делят компоненты на модули, которые изолируют код и позволяют его переиспользовать.

Для реализации SPA паттерна будет использоваться фреймворк Vue – это популярный, современный и открытый JavaScript-фреймворк для построения масштабируемого и отзывчивого пользовательского интерфейса. Его отличительной особенностью является низкий входной порог и необычайная простота использования, что делает этот фреймворк хорошим выбором для разработчиков любого уровня и опыта.

3.3 Разработка физической модели данных

Физическая модель данных представляет собой реализацию логической модели данных на уровне базы данных в рамках тестового веб-сервера [6, 22, 24]. Модели данных на веб-клиенте будут практически полностью идентичны таблицам БД за некоторыми исключениями, поэтому они будут опущены.

3.3.1 Список таблиц базы данных

В этом пункте представлен минимальный набор таблиц базы данных, необходимый для реализации минимально жизнеспособного продукта (MVP, Minimum Valuable Product), которого будет достаточно для того, чтобы покупатели могли ознакомиться с каталогом товаров и информацией о

товарах, добавить желаемые товары в корзину для оформления заказа и впоследствии оформить заказ через веб-сайт.

В таблице 1 представлен перечень таблиц базы данных, необходимый для функционирования MVP варианта ИС.

Таблица 1 – Таблицы базы данных

Наименование	Описание
customers	Содержит данные о покупателях
customers_to_orders	Содержит связки «идентификатор покупателя» к «идентификатор заказа»
products	Содержит данные о товарах
products_to_images	Содержит связки «идентификатор товара» к «ссылка на изображение товара»
categories	Содержит данные о категориях
categories_to_products	Содержит связки «идентификатор категории» к «идентификатор товара»
categories_to_categories	Содержит связки «идентификатор категории» к «идентификатор категории»
orders	Содержит данные о заказах
orders_to_products	Содержит связки «идентификатор заказа» к «идентификатор товара», а также кол-во товара
delivery_addresses	Содержит адреса доставки
delivery_shops	Содержит данные о магазинах для доставки/самовывоза

Каждая таблица имеет свой уникальный идентификатор (primary key), который связывает записи в разных таблицах.

3.3.2 Таблицы базы данных

1) Таблица базы данных customers.

С наполнением таблицы БД customers можно ознакомиться в таблице 2.

Таблица 2 – Таблица базы данных customers

Наименование	Тип	Описание
id	INT	Уникальный идентификатор покупателя
name	VARCHAR	Имя покупателя
email	VARCHAR	Адрес электронной почты покупателя
phone	VARCHAR	Номер телефона покупателя

SQL-запрос для создания таблицы представлен в листинге 1 [13, 20].

Листинг 1 – SQL-запрос для создания таблицы customers

```
CREATE TABLE customers(
    id INT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    phone VARCHAR(255) NOT NULL,
    email VARCHAR(255));
```

SQL-запрос для добавления покупателей представлен в листинге 2.

Листинг 2 – SQL-запрос для добавления покупателей в таблицу customers

```
INSERT INTO customers(name, phone, email)
VALUES
    („Валерий“, „79991112233“, NULL),
    („Дмитрий“, „79992223344“, „dmitry@mail.to“),
    („Андрей“, „71115554433“, NULL);
```

SQL-запрос для обновления информации представлен в листинге 3.

Листинг 3 – SQL-запрос для обновления информации в таблице customers

```
UPDATE customers SET name = „Никита“ WHERE id = 123;
```

SQL-запрос для удаления покупателя представлен в листинге 4.

Листинг 4 – SQL-запрос для удаления покупателя из таблицы customers

```
DELETE FROM customers WHERE id = 13;
```

2) Таблица базы данных customers_to_orders.

С наполнением таблицы БД customers_to_orders можно ознакомиться в таблице 3.

Таблица 3 – Таблица базы данных customers_to_orders

Наименование	Тип	Описание
binding_id	INT	Уникальный идентификатор связки
customer_id	INT	Уникальный идентификатор покупателя
order_id	INT	Уникальный идентификатор заказа

SQL-запрос для создания таблицы представлен в листинге 5.

Листинг 5 – SQL-запрос для создания таблицы customers_to_orders

```
CREATE TABLE customers_to_orders(
    binding_id INT PRIMARY KEY,
    customer_id INT NOT NULL,
    order_id INT NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES customers(id),
    FOREIGN KEY (order_id) REFERENCES orders(id));
```

SQL-запрос для добавления информации представлен на листинге 6.

Листинг 6 – SQL-запрос для добавления информации в таблицу customers_to_orders

```
INSERT INTO customers_to_orders (customer_id, order_id)
VALUES (123, 321);
```

3) Таблица базы данных products.

С наполнением таблицы БД products можно ознакомиться в таблице 4.

Таблица 4 – Таблица базы данных products

Наименование	Тип	Описание
id	INT	Уникальный идентификатор товара
name	VARCHAR	Наименование товара
description_short	TEXT	Короткое описание товара, отображаемое в каталоге товаров
description_long	TEXT	Полное описание товара, отображаемое в деталях товаров
image_preview_url	TEXT	Ссылка на изображение, отображаемое в каталоге товаров
price	VARCHAR	Цена товара

SQL-запрос для создания таблицы представлен в листинге 7.

Листинг 7 – SQL-запрос для создания таблицы products

```
CREATE TABLE products(  
  id INT PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  description_short TEXT NOT NULL,  
  description_long TEXT NOT NULL,  
  image_preview_url TEXT,  
  price VARCHAR NOT NULL);
```

SQL-запрос для добавления товара представлен в листинге 8.

Листинг 8 — SQL-запрос для добавления товаров в таблицу products

```
INSERT INTO products (name, description_short, description_long, image_preview_url, price)  
VALUES („Молоко“, „Объем 1л“, „Молоко Домик в Селе объемом 1л“,  
„https://static.images.ru/preview/someimage.png“, „39.99“);
```

SQL-запрос для обновления цены товара представлен на листинге 9.

Листинг 9 – SQL-запрос для обновления цены товара в таблице products

```
UPDATE products SET price = „49.99“ WHERE id = 123;
```

4) Таблица базы данных products_to_images.

С таблицей БД products_to_images можно ознакомиться в таблице 5.

Таблица 5 – Таблица базы данных products_to_images

Наименование	Тип	Описание
binding_id	INT	Уникальный идентификатор связки
product_id	INT	Уникальный идентификатор товара
image_url	TEXT	Ссылка на изображение товара

SQL-запрос для создания таблицы представлен в листинге 10.

Листинг 10 – Создание таблицы базы данных products_to_images

```
CREATE TABLE products_to_images(  
  binding_id INT PRIMARY KEY,  
  product_id INT NOT NULL,  
  image_url TEXT NOT NULL,  
  FOREIGN KEY (product_id) REFERENCES products(id));
```

5) Таблица базы данных categories.

С таблицей БД categories можно ознакомиться в таблице 6.

Таблица 6 – Таблица базы данных categories

Наименование	Тип	Описание
id	INT	Уникальный идентификатор категории
name	VARCHAR	Наименование категории

SQL-запрос для создания таблицы представлен в листинге 11.

Листинг 11 – Создание таблицы базы данных categories

```
CREATE TABLE categories(  
  id INT PRIMARY KEY,  
  name VARCHAR NOT NULL);
```

6) Таблица базы данных categories_to_products.

С таблицей БД categories_to_products можно ознакомиться в таблице 7.

Таблица 7 – Таблица базы данных categories_to_products

Наименование	Тип	Описание
binding_id	INT	Уникальный идентификатор связки
category_id	INT	Уникальный идентификатор категории
product_id	INT	Уникальный идентификатор товара

SQL-запрос для создания таблицы представлен в листинге 12.

Листинг 12 – Создание таблицы базы данных categories_to_products

```
CREATE TABLE categories_to_products(  
  binding_id INT PRIMARY KEY,  
  category_id INT NOT NULL,  
  product_id INT NOT NULL,  
  FOREIGN KEY (category_id) REFERENCES categories(id),  
  FOREIGN KEY (product_id) REFERENCES products(id));
```

7) Таблица базы данных categories_to_categories.

С таблицей БД categories_to_categories можно ознакомиться в таблице 8.

Таблица 8 – Таблица базы данных categories_to_categories

Наименование	Тип	Описание
binding_id	INT	Уникальный идентификатор связи
parent_category_id	INT	Уникальный идентификатор категории-родителя
child_category_id	INT	Уникальный идентификатор категории-ребенка

SQL-запрос для создания таблицы представлен в листинге 13.

Листинг 13 – Создание таблицы базы данных categories_to_categories

```
CREATE TABLE categories_to_categories(
    binding_id INT PRIMARY KEY,
    parent_category_id INT NOT NULL,
    child_category_id INT NOT NULL,
    FOREIGN KEY (parent_category_id) REFERENCES categories(id),
    FOREIGN KEY (child_category_id) REFERENCES categories(id));
```

8) Таблица базы данных orders.

С таблицей БД orders можно ознакомиться в таблице 9.

Таблица 9 – Таблица базы данных orders

Наименование	Тип	Описание
id	INT	Уникальный идентификатор заказа
customer_id	INT	Уникальный идентификатор покупателя
created_at	DATETIME	Дата и время создания заказа
updated_at	DATETIME	Дата и время последнего обновления заказа
status	ENUM	Статус заказа: CREATED; AWAITING_PAYMENT; ASSEMBLING; DELIVERING; AWAITING_PICK_UP; DONE; CANCELLED.

Продолжение таблицы 9

Наименование	Тип	Описание
payment_type	ENUM	Тип оплаты: ONLINE; CASH_ON_DELIVERY; CARD_ON_DELIVERY; CASH_ON_PICK_UP; CARD_ON_PICK_UP; ANY_ON_PICK_UP.
total_price	VARCHAR	Суммарная стоимость заказа
delivery_price	VARCHAR	Стоимость доставки заказа
is_payed	BIT	Флаг, индицирующий состояние оплаты
approx_delivery_date	TEXT	Человеко-читаемое примерное время доставки заказа для покупателя
delivery_type	ENUM	Тип доставки (DELIVERY, PICK_UP)
delivery_address_id	INT	Уникальный идентификатор адреса доставки (NOT NULL только если deliveryType == DELIVERY)
delivery_shop_id	INT	Уникальный идентификатор магазина доставки (NOT NULL только если deliveryType == PICK_UP)

SQL-запрос для создания таблицы представлен в листинге 14.

Листинг 14 – Создание таблицы базы данных orders

```
CREATE TABLE orders(
    id INT PRIMARY KEY,
    customer_id INT NOT NULL,
    created_at DATETIME NOT NULL,
    updated_at DATETIME NOT NULL,
    status ENUM(„CREATED“, „AWAITING_PAYMENT“, „ASSEMBLING“,
„DELIVERING“, „AWAITING_PICK_UP“, „DONE“, „CANCELLED“) NOT NULL,
    payment_type ENUM(„ONLINE“, „CASH_ON_DELIVERY“,
„CARD_ON_DELIVERY“, „CASH_ON_PICK_UP“, „CARD_ON_PICK_UP“,
„ANY_ON_PICK_UP“) NOT NULL,
    total_price VARCHAR(255) NOT NULL,
    delivery_price VARCHAR(255) NOT NULL,
    is_payed BIT(1) NOT NULL,
    approx_delivery_date TEXT,
    delivery_type ENUM(„DELIVERY“, „PICK_UP“) NOT NULL,
    delivery_address_id INT,
    delivery_shop_id INT,
    FOREIGN KEY (delivery_address_id) REFERENCES delivery_addresses(id),
```

FOREIGN KEY (delivery_shop_id) REFERENCES shops(id));

9) Таблица базы данных orders_to_products.

С таблицей БД orders_to_products можно ознакомиться в таблице 10.

Таблица 10 – Таблица базы данных orders_to_products

Наименование	Тип	Описание
binding_id	INT	Уникальный идентификатор связки
order_id	INT	Уникальный идентификатор заказа
product_id	INT	Уникальный идентификатор товара
amount	INT	Количество товара, находящегося в заказе

SQL-запрос для создания таблицы представлен в листинге 15.

Листинг 15 – Создание таблицы базы данных orders_to_products

```
CREATE TABLE orders_to_products(  
    binding_id INT PRIMARY KEY,  
    order_id INT NOT NULL,  
    product_id INT NOT NULL,  
    amount INT NOT NULL,  
    FOREIGN KEY (order_id) REFERENCES orders(id),  
    FOREIGN KEY (product_id) REFERENCES products(id));
```

10) Таблица базы данных delivery_addresses.

С таблицей БД delivery_addresses можно ознакомиться в таблице 11.

Таблица 11 – Таблица базы данных delivery_addresses

Наименование	Тип	Описание
id	INT	Уникальный идентификатор адреса доставки
city	VARCHAR	Наименование города
street	VARCHAR	Наименование улицы
house	VARCHAR	Номер дома
apartment	VARCHAR	Номер квартиры
commentary	TEXT	Комментарий от покупателя для курьеров

SQL-запрос для создания таблицы представлен в листинге 16.

Листинг 16 – Создание таблицы базы данных delivery_addresses

```
CREATE TABLE delivery_addresses(  
  id INT PRIMARY KEY,  
  city VARCHAR(255) NOT NULL,  
  street VARCHAR(255) NOT NULL,  
  house VARCHAR(255) NOT NULL,  
  apartment VARCHAR(255),  
  commentary TEXT);
```

11) Таблица базы данных shops.

С таблицей БД shops можно ознакомиться в таблице 12.

Таблица 12 – Таблица базы данных shops

Наименование	Тип	Описание
id	INT	Уникальный идентификатор магазина
address	TEXT	Полный адрес магазина

SQL-запрос для создания таблицы представлен в листинге 17.

Листинг 17 – Создание таблицы базы данных shops

```
CREATE TABLE shops(id INT PRIMARY KEY, address TEXT NOT NULL);
```

3.4 Контрольный пример реализации проекта

Для реализации веб-представительства использовалась коллекция инструментов, библиотек и фреймворков, из которых можно выделить:

- Vue – для создания пользовательского интерфейса и разбиения веб-сайта на переиспользуемые и изолированные компоненты;
- Nuxt – для упрощения рутинных активностей в процессе разработки на Vue, а также для расширения доступного инструментария;
- Tailwindcss – для упрощения стилизации компонентов.

Разработка была начата из специального шаблона Vue+Nuxt для быстрого старта, в котором готова структура проекта, написаны конфигурационные файлы, созданы базовые стили и компоненты.

Тело приложения содержит код, представленный на листинге 18.

Листинг 18 – Точка входа в веб-приложение

```
<template>
  <div>
    <VmHeaderInfo />
    <VmHeader />
    <main>
      <nuxt class="flex-1 flex-col max-w-7xl m-auto" />
      <VmLoginModal />
      <VmSignupModal />
      <VmCheckoutModal />
    </main>
    <VmFooter />
  </div>
</template>

<script>
import VmHeaderInfo from '@/components/header/HeaderInfo';
import VmHeader from '@/components/header/Header';
import VmFooter from '@/components/footer/Footer';
import VmLoginModal from '@/components/modal/Login';
import VmSignupModal from '@/components/modal/Signup';
import VmCheckoutModal from '@/components/modal/Checkout';

export default {
  components: {
    VmHeaderInfo,
    VmHeader,
    VmFooter,
    VmLoginModal,
    VmSignupModal,
    VmCheckoutModal
  }
}
</script>

<style lang="scss">
body {
  @apply flex;
  @apply flex-col;
  height: 100vh;
  margin: 0;
}
```

```

}

.input {
  @apply block;
  @apply w-full;
  @apply px-4;
  @apply py-2;
  @apply text-xl;
  @apply font-normal;
  @apply bg-grey_light;
  @apply bg-clip-padding;
  @apply border;
  @apply border-solid;
  @apply border-grey_light;
  @apply rounded;
  @apply transition;
  @apply ease-in-out;
  @apply m-0;
  @apply focus:border-black;
  @apply focus:outline-none;
}

.modal {
  @apply top-0;
  @apply right-0;
  @apply left-0;
  @apply bottom-0;
  @apply z-50;
  @apply items-center;
  @apply justify-center;
}

.modal-background {
  @apply bg-grey_dark/80;
  @apply w-full;
  @apply h-full;
  @apply z-10;
  @apply fixed;
  @apply top-0;
}

.modal-wrapper {
  @apply bg-white;
  @apply z-20;
  @apply rounded-2xl;
  @apply w-96;
}
</style>

```

Представленный выше листинг кода является точкой входа в приложение и содержит те компоненты приложения, которые всегда отображаются пользователю. Компонент `nuxt` с помощью кодогенерации и различных трансформаций динамически, во время работы приложения подменяется на конечную страницу, которая соответствует текущему местонахождению пользователя в приложении.

На листинге 19 представлен код главной страницы веб-сайта, которая отображается при посещении веб-сайта впервые.

Листинг 19 – Главная страница приложения

```
<template>
  <div>
    <VmHero />
    <VmSubheader />
    <VmProductsList />
  </div>
</template>

<script>
import VmProductsList from '@components/products_list/ProductsListContainer';
import VmHero from '@components/hero/Hero';
import VmSubheader from '@components/subheader/Subheader'

export default {
  name: 'index',
  components: {
    VmProductsList,
    VmHero,
    VmSubheader
  }
};
</script>
```

На рисунках 14, 15 и 16 представлены скриншоты пользовательского интерфейса главной страницы, в случае, если пользователь пользуется веб-сайтом с персонального компьютера.

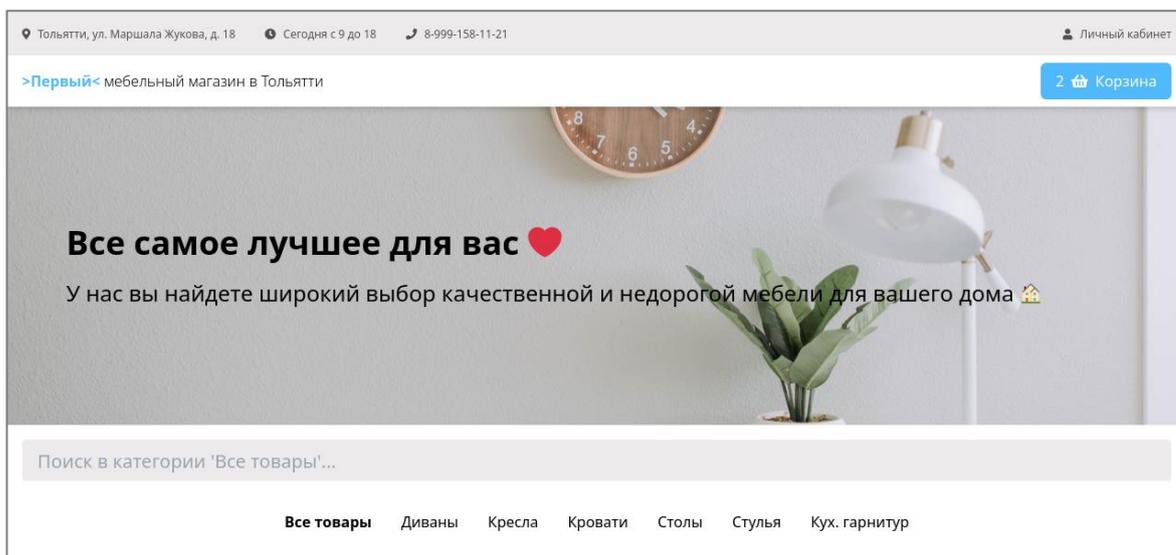


Рисунок 14 – Скриншот главной страницы веб-сайта с ПК, демонстрирующий заголовок

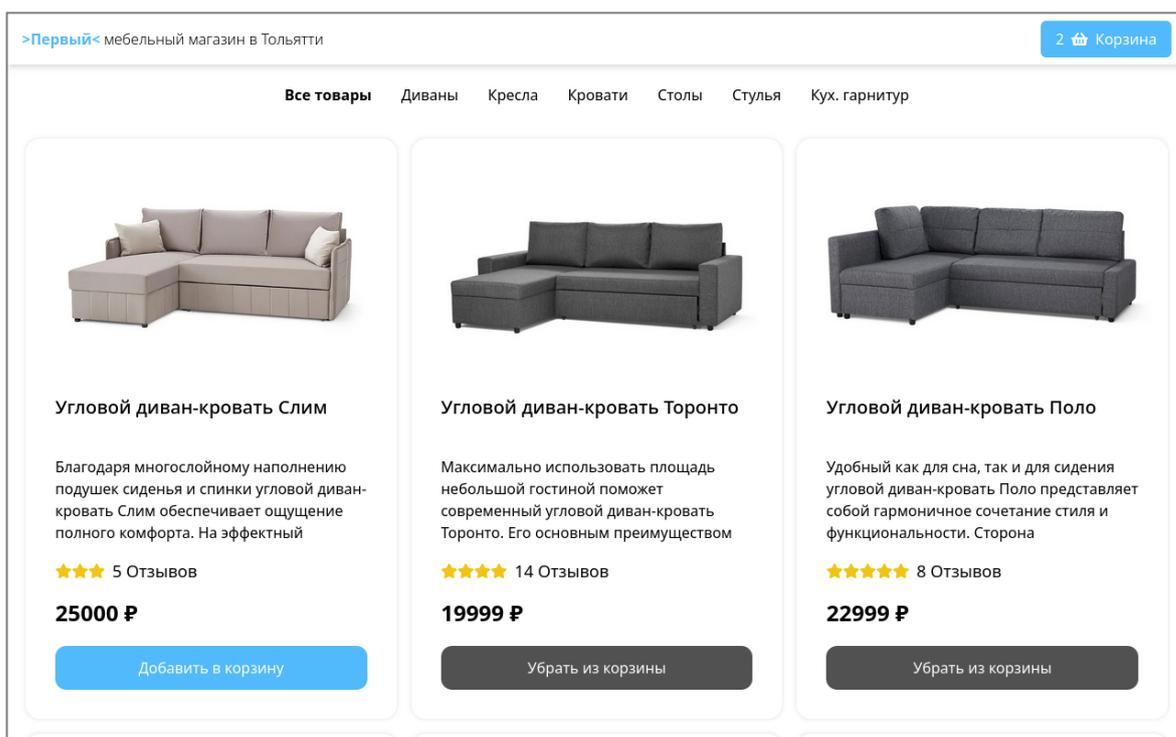


Рисунок 15 – Скриншот главной страницы веб-сайта с ПК, демонстрирующий каталог товаров

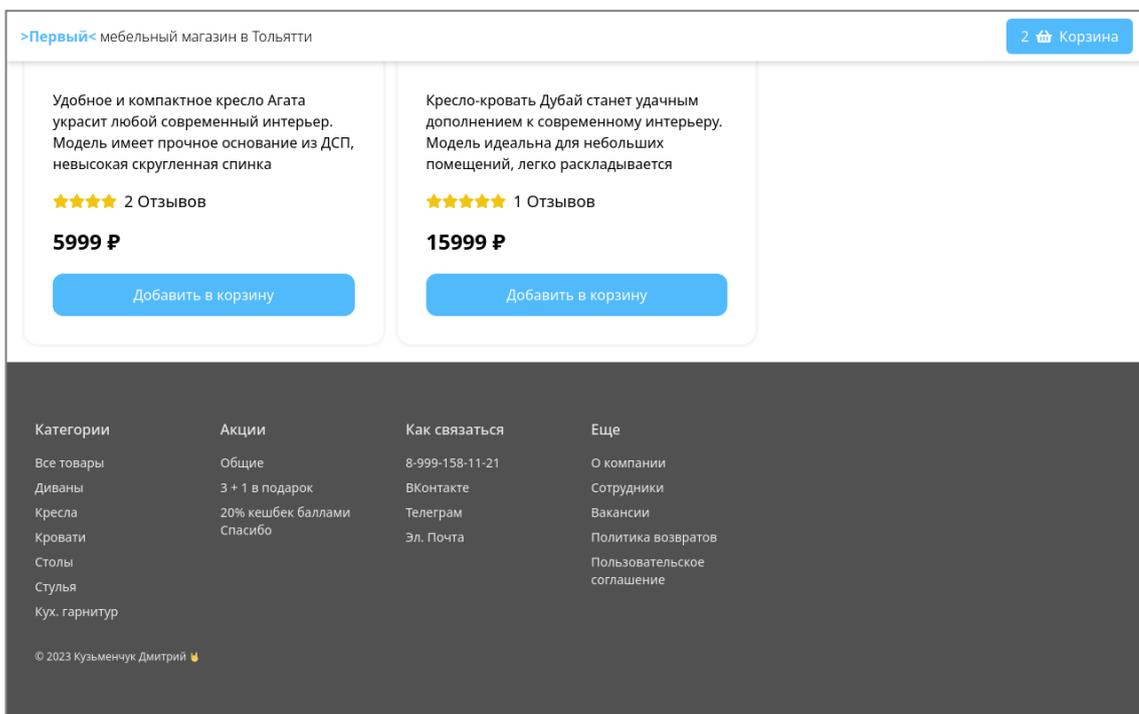


Рисунок 16 – Скриншот главной страницы веб-сайта с ПК, демонстрирующий Footer

На рисунке 17 представлены скриншоты пользовательского интерфейса главной страницы, в случае, если пользователь пользуется мобильным устройством.

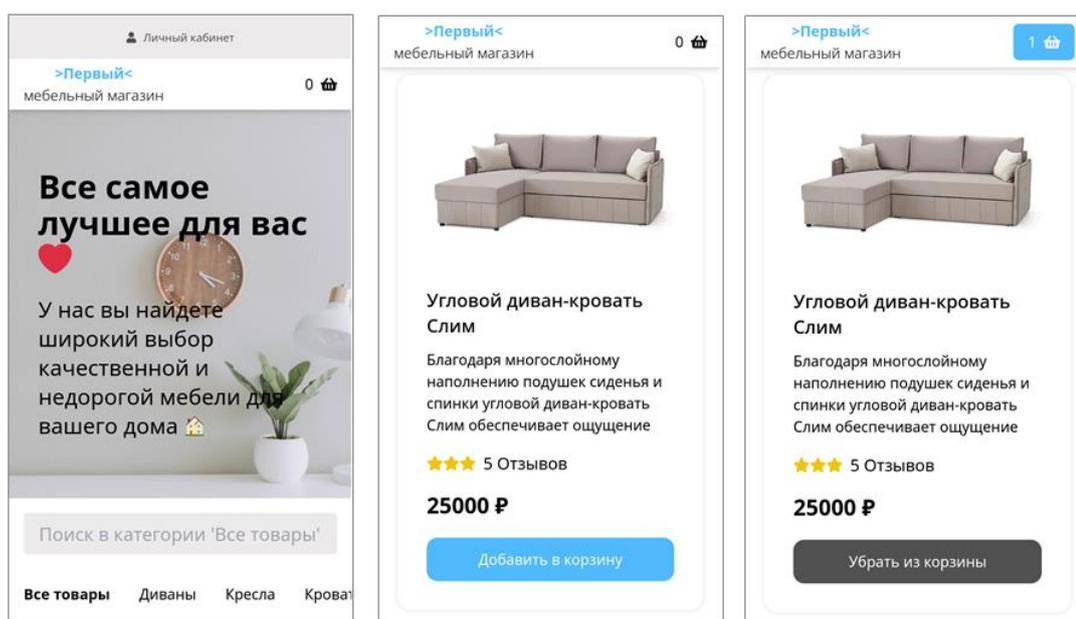


Рисунок 17 – Скриншоты главной страницы веб-сайта с мобильного устройства, демонстрирующие заголовок и каталог товаров

На рисунке 18 представлены скриншоты корзины товаров при просмотре её с мобильного устройства.

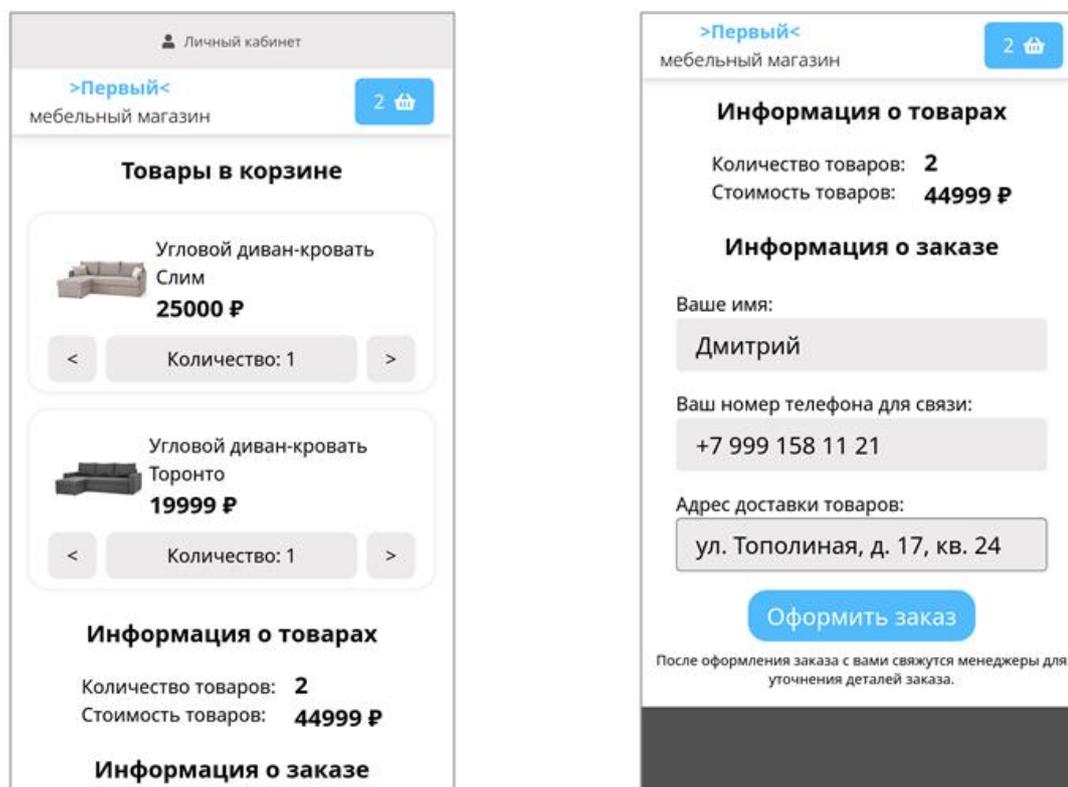


Рисунок 18 – Скриншоты корзины товаров с мобильного устройства

Разработанная выше функциональность достаточна для запуска MVP-версии приложения, которое позволит как можно быстрее получать заказы и обратную связь от покупателей.

3.5 Оценка экономической эффективности проекта

Для оценки экономической эффективности по разработке и внедрению АИС на примере интернет-магазина мебели необходимо решить следующие задачи [16]:

- расчет затрат на разработку интернет-магазина;
- расчет эксплуатационных затрат за год;

- расчет экономической эффективности капитальных затрат;
- расчет срока окупаемости.

Очевидные плюсы создания интернет-магазина – высокая эффективность при сравнительно небольшой стоимости, что позволяет компании увеличить базу клиентов, повысить популярность бренда и укрепиться на рынке, и все это приносит владельцам реальную прибыль.

Поставим основные задачи для интернет-магазина:

- увеличение объема продаж на 10% за счет привлечения новых клиентов;
- расширение способов продвижения продукции (за счет использования новых способов продвижения).

Для создания, развития и сопровождения интернет-магазина будет увеличен штат сотрудников предприятия – двое новых сотрудников:

- fullstack frontend разработчик;
- backend for frontend разработчик.

Единовременные затраты на создание интернет-магазина включают следующие статьи затрат:

- затраты на приобретение и установку оборудования;
- затраты на разработку интернет-магазина.

Рассчитаем зарплату для разработчиков, приведя результаты в таблице 13.

Таблица 13 – Расчет затрат на разработку сайта

Сотрудник	Количество рабочих дней (в мес.)	З./п. за рабочий день, тыс. руб. (8 ч)	З./п. за месяц, тыс. руб.
Frontend	20	2,5	50
Backend	20	2,5	50

Средний срок разработки интернет-магазина – 5 месяцев. Рассчитаем затраты на разработку (Тр) с учетом этого срока:

$$Тр = (50 * 2) * 5 = 500 \text{ тыс. руб.}$$

Расчет затрат на приобретение оборудования производим с учетом двух новых разработчиков (таблица 14).

Таблица 14 – Стоимость оборудования для новых разработчиков интернет-магазина

Наименование	Стоимость, тыс. руб.
Компьютер	60 (x2)
Монитор	20 (x2)
Периферийные устройства	2.5 (x2)
Итого	165

Затраты на установку и монтаж оборудования (УМО) определим по общепринятым нормативам (2,5%), в процентах от стоимости технических и программных средств (ТПС).

$$УМО = ТПС * 2,5\%$$

$$УМО = 165 * 2.5\% = 4,125 \text{ тыс. руб.}$$

Рассчитаем норму амортизационных отчислений линейным способом, если первоначальная стоимость оборудования составляет 165 тыс. руб., а срок полезного использования (СПИ) компьютерной техники составляет до трех лет.

$$СПИ = \text{Стоимость оборудования} / \text{СПИ}$$

$$СПИ = 165 / 36 = 4,6 \text{ тыс. руб.}$$

$$\text{Амортиз.} = СПИ * \text{Срок эксплуатации}$$

$$\text{Амортиз.} = 4,6 * 5 = 23 \text{ тыс. руб.}$$

Единовременные затраты на разработку интернет-магазина сведем в таблицу 15.

Таблица 15 – Единовременные затраты на разработку интернет-магазина

Статья затрат	Сумма, тыс. руб.
Затраты на приобретение оборудования	165
Амортизация приобретенного оборудования	23
Затраты на установку оборудования	4,125
Затраты на зарплаты разработчиков	500
Итого	692,125

Расчеты показали, что единовременные вложения за 5 месяцев в сумме составят 692, 125 тыс. руб.

Далее рассчитаем эксплуатационные затраты за год. Они представляют собой сумму затрат связанных с эксплуатацией интернет-магазина и включают следующие статьи затрат:

- амортизационные отчисления;
- выплата заработной платы разработчикам;
- затраты на рекламную кампанию.

Амортизация уже была посчитана выше по тексту. Заработная плата уменьшится вдвое, т.к. активной разработкой и поддержкой интернет-магазина будет заниматься один разработчик. Расходы на рекламную кампанию по предварительным расчетам составят от 103,2 тыс. руб. в год, детализация в таблице 16.

Таблица 16 – Затраты на рекламную кампанию за год

Рекламная площадка	Цена в месяц, руб.	Цена в год, руб.
Реклама в Яндекс.Директ	3 000	36 000
Реклама в Яндекс.Маркет	2 900	34 800
Реклама в VK	2 700	32 400
Итого	8 600	103 200

Сведем расчеты эксплуатационных затрат в таблицу 17.

Таблица 17 – Сводные показатели по расчету эксплуатационных затрат за год

Наименование затрат	Стоимость, тыс. руб.
Амортизация оборудования	55,2
Затраты на оплату труда	600
Затраты на рекламу	103,2
Итого	758,4

Расчеты показали, что эксплуатационные затраты за год составят 758,4 тыс. руб.

Посчитаем затраты в первый год разработки и поддержки интернет-магазина:

$$Тп = (758,4 / 12 * 7) + 692,125 = 1134,525$$

Далее проведем расчет эффективности, учитывая, что создание интернет-магазина позволит получить прирост выручки в размере 10% от годового оборота предприятия. Примем за исходную точку доход в размере 10000 тыс. руб., который необходимо улучшить.

Прибыль отчетного года определяется по формуле:

$$Пог = ДО - НК - З,$$

где ДО – доходы от основной деятельности, тыс. руб.;

НК – налоги и отчисления из выручки, тыс. руб.;

З – текущие затраты, тыс. руб.

Налог на добавленную стоимость НДС рассчитывается в размере 18% от общей суммы доходов от основной деятельности по формуле:

$$НДС = ДО * 0,18$$

Налог на добавленную стоимость будет равен

$$НДС = 10000 * 0,18 = 1800 \text{ тыс. руб.}$$

Теперь можно вычислить прибыль отчетного года:

$$Пог = 10000 - 1800 - 1134,525 = 7065,475$$

Экономический эффект от создания интернет-магазина выступает в виде чистой прибыли и определяется как разница между прибылью отчетного года и налогом на прибыль по формуле:

$$Пч = Пог - НП,$$

где НП – налог на прибыль (20%), тыс. руб.

$$НП = Пог * 0,2$$

$$НП = 7065,475 * 0,2 = 1413,095 \text{ тыс. руб.}$$

$$Пч = 7\ 065,475 - 1\ 413,095 = 5652,38 \text{ тыс. руб.}$$

Чистая прибыль предприятия в первый год составит 5652,38 тыс. руб.

Считаем, что к этой чистой прибыли прибавляется доход от интернет-магазина в объеме 10%.

$$Пчм = (Пч / 100 * 110) - Пч$$

$$Пчм = 6\ 217,618 - 5\ 652,38 = 562,238$$

Для упрощения будем считать, что каждый год чистая прибыль интернет-магазина увеличивается на 20% от чистой прибыли прошло года. Составим таблицу для вычисления периода окупаемости интернет-магазина, результат в таблице 18.

Таблица 18 – Расчеты окупаемости интернет-магазина

Период (год)	Инвестиции, тыс. руб.	Годовой доход, тыс. руб.	Годовые издержки, тыс. руб.	Денежный поток, тыс. руб.	Денежный поток нарастающим итогом, тыс. руб.
1	-692,125	562,238	-758,4	-196,162	-196,162
2	0	674,686	-758,4	-83,714	-279,876
3	0	809,623	-758,4	51,223	-228,653
4	0	971,548	-758,4	213,148	-15,505
5	0	1 165,858	-758,4	407,458	391,953

Исходя из информации, отображенной в таблице 18, можно сделать вывод, что интернет-магазин окупится к пятому году при условии, что он в

первый год принесет 10% прибыли от прибыли предприятия и далее будет улучшать этот показатель на 20% в сравнении с прошлым годом.

Выводы по главе 3

В рамках этой главы разработана системная архитектура проекта с применением UML-диаграммы классов, тщательно обоснованы и выбраны программные средства для реализации веб-представительства, выполнена разработка физических моделей данных проекта, представлен контрольный пример реализации проекта, включающий в себя исходные коды и скриншоты реализованной ИС, а также выполнено экономическое обоснование проекта.

Заключение

На основании проведенного исследования можно сделать вывод о важности веб-представительства для предприятий в сфере оптово-розничной торговли. В современном мире все больше людей предпочитают совершать покупки онлайн, наличие привлекательного и функционального веб-сайта становится необходимостью для успешного бизнеса.

В процессе работы была дана характеристика деятельности предприятия оптово-розничной торговли, разработаны функциональные модели бизнес-процессов AS-IS и TO-BE, а также сформулированы требования к автоматизированному решению. Описана логическая модель веб-представительства, разработана логическая модель данных, а также сформулированы требования к аппаратному и программному обеспечению. Физическая модель данных была разработана, представлен контрольный пример проектного решения, и проведена оценка экономической эффективности проекта.

В результате работы были получены ценные рекомендации и разработанное веб-представительство, которое позволит предприятию оптово-розничной торговли улучшить свои показатели продаж и обслуживания клиентов. Это представляет значительный потенциал для дальнейшего развития и успеха предприятия на рынке.

Подводя итог: разработка веб-представительства является важным шагом для предприятия оптово-розничной торговли, позволяющим следовать современным тенденциям и удовлетворять потребности клиентов. Реализация данного проекта способствует повышению конкурентоспособности предприятия и созданию благоприятных условий для его развития в долгосрочной перспективе.

Список используемой литературы

1. Агальцов В.П. Базы данных: учеб. для вузов по направлению 09.03.01 «Информатика и вычисл. техника» Кн. 2 Распределенные и удаленные базы данных / В.П. Агальцов. – Документ Bookread2. – М. : ФОРУМ [и др.], 2019. – 270 с.
2. Бизнес-планирование : учебник / Л. В. Бобков [и др.] под ред. Т. Г. Попадюк, В. Я. Горфинкеля. – Документ Bookread2. – М. : Вузов. учеб. [и др.], 2018. – 295 с.
3. Буч Г. Введение в UML от создателей языка. 2-е изд. : пер. с англ. Мухин Н. / Г. Буч, Д. Рамбо, И. Якобсон. – Москва : ДМК Пресс, 2012. – 494 с.
4. Варфоломеева А. О. Информационные системы предприятия : учеб. пособие для вузов по направлению 09.03.03. "Приклад. информатика" и др. экон. специальностям / А. О. Варфоломеева, А. В. Коряковский, В. П. Романов. – 2-е изд., перераб. и доп. – Документ Bookread2. – М. : ИНФРА-М, 2019. – 330 с.
5. Гвоздева В. А. Информатика, автоматизированные информационные технологии и системы: учебник / В. А. Гвоздева. Москва: Форум: Инфра-М, 2019. – 541 с.
6. Голицына О. Л. Базы данных : учеб. пособие для студентов вузов по направлению 09.03.03 "Приклад. информатика" / О. Л. Голицына, Н. В. Максимов, И. И. Попов. – 4-е изд., перераб. и доп. – Документ Bookread2. – М. : Форум [и др.], 2019. – 399 с.
7. Документация по веб-фреймворку Ktor на англ. [Электронный ресурс] – URL: <https://ktor.io/docs/welcome.html>.
8. Документация по языку программирования Kotlin на англ. [Электронный ресурс] – URL: <https://kotlinlang.org/docs/home.html>.

9. Душин В.К. Теоретические основы информационных процессов и систем : учеб. для вузов по направлению "Информ. системы" и по специальностям "Информ. системы и технологии", "Сервис БРЭА", "Информ. сервис", "Сервис компьютерной и микропроцессорной техники", "Сервис" / В. К. Душин. – 5-е изд. - Документ HTML. – М. : Дашков и К, 2019. – 348 с.
10. Жемеров Д. Kotlin в действии : пер. с англ. Киселев А. Н. / Д. Жемеров, С. Исакова. – Москва : ДМК Пресс, 2018. – 402 с.
11. Ипатова Э. Р. Методологии и технологии системного проектирования информационных систем : учебник / Э. Р. Ипатова, Ю. В. Ипатов. – 2-е изд., стер. – Москва : ФЛИНТА, 2018. – 257 с.
12. Леоненков А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose [Электронный ресурс] : учебное пособие. М. : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. 317 с. URL: <https://www.iprbookshop.ru/97554.html> (дата обращения: 10.03.2023).
13. Маркин А. В. Построение запросов и программирование на SQL : учеб. пособие для студентов вузов по направлению "Информ. системы и технологии" / А. В. Маркин. – Изд. 3-е, перераб. и доп. – М. : Диалог-МИФИ, 2016. – 384 с.
14. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения – Санкт-Петербург : Питер, 2021. – 352 с.
15. Мартин Р. Чистый код: создание, анализ и рефакторинг. – Санкт-Петербург : Питер, 2021. – 464 с.
16. Мкртычев С. В. Прикладная информатика. Бакалаврская работа : электрон. учеб.-метод. пособие / С.В. Мкртычев, О.М. Гущина, А.В. Очеповский. – Тольятти : Изд-во ТГУ, 2019. – 74 с.

17. Олифер В. Компьютерные сети. Принципы, технологии, протоколы : Юбилейное издание. / В. Олифер, Н. Олифер. – Санкт-Петербург : Питер, 2021. – 1008 с.
18. Основы автоматизированного проектирования : учеб. для вузов по направлению 09.03.01 "Информатика и вычисл. техника" (квалификация (степень) "бакалавр") / А. Н. Божко [и др.] под ред. А. П. Карпенко. - Документ Bookread2. – М. : ИНФРА-М, 2018. – 345 с.
19. Пирогов В. Ю. Информационные системы и базы данных. Организация и проектирование / В. Ю. Пирогов. – СПб.: «БХВ-Петербург», 2020. – 254 с.
20. Шилдс У. SQL: быстрое погружение – Санкт-Петербург : Питер, 2022. – 224 с.
21. An Introduction to the Unified Modeling Language [Электронный ресурс]. URL: https://personalpages.bradley.edu/~young/CS592M120_OLD/handoutUML.pdf (дата обращения: 10.03.2023).
22. Database Systems Concepts, Languages and Architectures [Электронный ресурс]. URL: <https://users.dimi.uniud.it/~angelo.montanari/logicDesign.pdf> (дата обращения: 10.03.2023).
23. Martin Kleppmann. Designing Data-Intensive Applications : O'Reilly Media, 2018.
24. Sasaki B., Chao J., Howard R.. Graph Databases For Beginners : Self-publishing, 2018.
25. Sequence Diagram [Электронный ресурс]. URL: <https://www.smartdraw.com/sequence-diagram/> (дата обращения: 10.03.2023).