

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

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Разработка программного обеспечения

(направленность (профиль)/специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка и администрирование веб-приложения внутренней логистики предприятия»

Обучающийся

А. О. Аллюсов

(И.О. Фамилия)

(личная подпись)

Руководитель

к.т.н, Н. В. Хрипунов

(ученая степень, звание, И.О. Фамилия)

Консультант

старший преподаватель, М. В. Дайнеко

(ученая степень, звание, И.О. Фамилия)

Аннотация

Бакалаврская работа выполнена на тему: «Разработка и администрирование веб – приложения внутренней логистики предприятия».

Выпускная квалификационная работа (ВКР) состоит из введения, трех глав, 20 таблиц, 32 рисунков, заключения и списка литературы из 22 источников.

Целью данной ВКР является разработка и внедрение веб-приложения, которое автоматизирует процессы внутренней логистики предприятия и повышает эффективность управления.

Первая глава содержит описание структуры и деятельности ООО «Волжский Картон». Приведены технико-экономические характеристики предприятия, рассмотрены существующие бизнес-процессы в нотации IDEF0 и предложена их оптимизация путем внедрения автоматизированной системы, веб - приложения.

Во второй главе представлено проектирование системы. Описаны основные компоненты программного обеспечения, приведена диаграмма прецедентов, состоящая из 4 акторов – кладовщик, инженер – снабженец, инженер – производитель и администратор. Приведены диаграммы классов и логическая модель данных, состоящие из 6 блоков – rawmaterial, product, request, warehouse, shipment, user.

Третья глава посвящена практической реализации системы. В ней рассмотрена системная архитектура проекта, состоящая из клиентской и серверной частей, подробно описана функциональность веб-приложения, а также приведены результаты тестирования.

В заключении хотелось бы подчеркнуть, что в ходе выполнения данной выпускной квалификационной работы было разработано веб – приложение для управления внутренней логистикой предприятия ООО «Волжский картон», что позволило автоматизировать ключевые бизнес-процессы предприятия и повысить их эффективность.

Abstract

The title of the graduation work is *Developing and administrating a web application for internal logistics of an enterprise*.

The graduation work consists of an introduction, 3 chapters, 32 figures, 20 tables, a conclusion, and a list of 22 references.

The aim of this graduation work is to develop and implement a web application that automates the internal logistics processes of an enterprise and improves the management efficiency.

The first chapter provides a description of the structure and the activity of the company under research. It presents the technical and economic characteristics of the company and examines the existing business processes using IDEF0 notation. Optimization of these processes is proposed through the implementation of an automated system – the web application.

The second chapter deals with designing the system. It describes the main components of the software and considers a use-case diagram consisting of four actors: storekeeper, supply engineer, production engineer, and administrator. It also presents class diagrams and a logical data model comprising six blocks: raw material, product, request, warehouse, shipment, and user.

The third chapter dwells on the practical implementation of the system. It examines the project system architecture consisting of client-side and server-side parts, as well as describes the functionality of the web application in detail, and provides the testing results.

In conclusion, it should be highlighted that the web application designed for managing the internal logistics of the company in question allows to automate its key business processes and significantly improves their efficiency.

Оглавление

Введение	5
Глава 1 Анализ предметной области ООО «Волжский Картон»	7
1.1 Описание организации, являющейся объектом исследования	7
1.2 Техничко – экономическая характеристика ООО «Волжский Картон»	7
1.3 Функциональная модель организации «как есть»	9
1.4 Функциональная модель организации «как должно быть»	14
1.5 Анализ лучших практик и существующих разработок в предметной области	20
1.6 Функциональные требования проекта	22
Глава 2 Архитектура проекта и особенности его реализации	25
2.1 Диаграмма прецедентов веб - приложения	25
2.2 Диаграмма классов разрабатываемого веб - приложения	28
2.3 Логическая модель данных веб - приложения	31
Глава 3 Физическое проектирование веб – приложения	37
3.1 Системная архитектура проекта	37
3.2 Описание функциональности веб – приложения	39
3.3 Тестирование веб – приложения	47
Заключение	49
Список используемой литературы	50

Введение

В условиях современного рынка и растущей конкуренции компаниям требуется внедрение эффективных информационных систем, которые способны повысить производительность, оптимизировать затраты и упростить управление ресурсами. Интеграция современных технологий в бизнес-процессы становится не просто преимуществом, а необходимостью для поддержания конкурентоспособности. ООО «Волжский картон» — компания, занимающаяся производством картонных изделий, — сталкивается с задачей модернизации процессов управления внутренней логистики. Внедрение автоматизированной системы позволит значительно упростить координацию между подразделениями, повысить прозрачность операций и сократить временные затраты на обработку данных.

Цель работы — разработка и внедрение веб-приложения, которое автоматизирует процессы внутренней логистики предприятия и повышает эффективность управления. Создание такой системы направлено на повышение эффективности работы предприятия, снижение операционных рисков, улучшение взаимодействия между различными подразделениями компании, а также оптимизацию управления материальными и информационными потоками.

Для достижения цели исследования были поставлены следующие задачи:

- изучить организационную структуру и бизнес-процессы предприятия, определить цели и задачи автоматизации;
- проанализировать лучшие практики и существующие решения в области управления внутренней логистики, чтобы выбрать подходящие методы и инструменты для разработки;
- разработать архитектуру системы, включающую логическую и физическую модели данных, которые отразят реальную структуру и процессы предприятия;

- спроектировать и реализовать веб-приложение, используя объектно-ориентированный подход и современные технологии, обеспечивающие удобство и гибкость интерфейса;
- провести тестирование системы, чтобы выявить возможные ошибки и недостатки.

Объектом исследования являются процессы внутренней логистики на предприятии ООО «Волжский картон».

Предмет исследования — веб-приложение для автоматизации логистических процессов на предприятии ООО «Волжский картон».

Внедрение автоматизированной системы внутренней логистики необходимо для повышения эффективности работы, сокращения временных затрат и улучшения координации между подразделениями ООО «Волжский картон». Это позволит предприятию повысить прозрачность процессов и конкурентоспособность на рынке. Это формирует актуальность темы данной работы.

Настоящая работа состоит из трёх глав, каждая из которых посвящена определённому этапу разработки системы. В первой главе проводится анализ текущего состояния предприятия, выявляются цели и задачи автоматизации. Вторая глава охватывает проектирование системы и построение архитектуры. В третьей главе рассматриваются вопросы разработки и тестирования.

Комплексный подход к автоматизации управления внутренней логистикой позволит ООО «Волжский картон» не только позволит повысить уровень контроля над процессами, но и значительно улучшить их эффективность, снизить затраты и повысить скорость выполнения задач. В долгосрочной перспективе внедрение данной системы будет способствовать повышению конкурентоспособности предприятия на рынке, улучшению качества предоставляемых услуг и созданию благоприятных условий для дальнейшего развития.

Глава 1 Анализ предметной области ООО «Волжский Картон»

1.1 Описание организации, являющейся объектом исследования

ООО «Волжский картон» — это производственное предприятие, основанное в 2013 году, с ведущими специалистами, которые работают в сфере гофрокартона. Основная цель компании — предоставление качественных услуг на рынке гофрокартона и картонной упаковки. Предприятие производит разнообразные виды упаковки из гофрокартона для широкого спектра отраслей промышленности, начиная от четырехклапанных коробов до сложных упаковочных решений, таких как гофроподдоны, лотки, и коробки под конкретные запросы клиентов.

Задачи и цели предприятия:

- обеспечение стабильного и высокого качества продукции, контролируемого на каждом этапе производственного процесса;
- гибкая ценовая политика и оптимизация расходов клиентов за счет комплексного подхода к производству упаковки;
- разработка индивидуальных решений для клиентов, начиная с консультаций и проектирования до доставки готовой продукции;
- поддержание бесперебойных поставок и соблюдение сроков исполнения заказов для удовлетворения потребностей клиентов;

1.2 Техничко – экономическая характеристика ООО «Волжский Картон»

Рассмотрю организационную структуру предприятия с распределением ответственности, полномочий и взаимоотношений между работниками предприятия [12]. Сделаю соответствующую схему, используя инструмент Draw.io (рисунок 1):

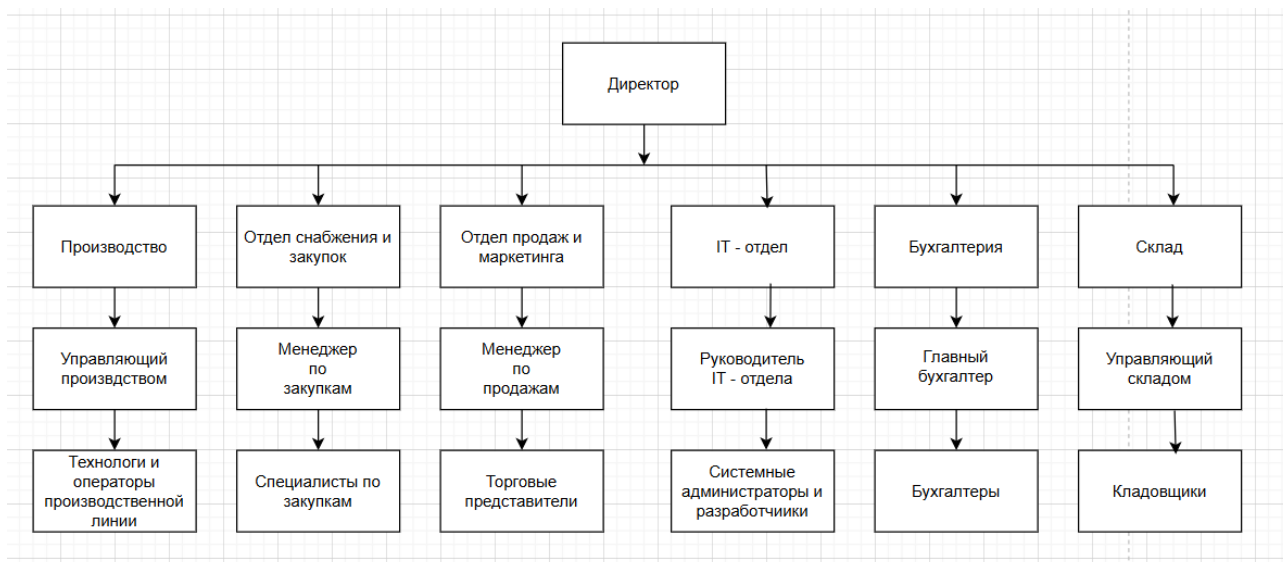


Рисунок 1 – Структура организации

Опишу обязанности. Директор руководит предприятием и принимает стратегические решения.

Управляющий производством отвечает за производство продукции и контроль качества. Технологи разрабатывают и контролируют технологические процессы. Операторы производственной линии выполняют производственные операции и контроль продукции.

Менеджер по закупкам отвечает за поставки сырья и управление складом сырья. Специалисты по закупкам занимаются поиском и закупкой сырья, контролируют поставки.

Менеджер по продажам привлекает клиентов и управляет продажами. Торговые представители привлекают клиентов и взаимодействуют с ними, осуществляют продажи.

Руководитель IT-отдела управляет IT-инфраструктурой, отвечает за техническую поддержку и безопасность. Системные администраторы поддерживают серверы и сети в рабочем состоянии. Разработчики разрабатывают и поддерживают программное обеспечение на предприятии.

Главный бухгалтер управляет финансами, ведет бухгалтерский учет и отчетность. Бухгалтеры занимаются учётом и ведут отчётность по финансам предприятия.

Управляющий складом распределяет задачи кладовщикам и координирует их работу. Кладовщики осуществляют приемку и размещение груза, подготовку и отгрузку груза.

1.3 Функциональная модель организации «как есть»

ООО «Волжский Картон» — это высокотехнологичное предприятие, специализирующееся на производстве упаковки из гофрокартона. С момента своего основания в 2013 году компания зарекомендовала себя как надежный поставщик качественной продукции, активно внедряя новые технологии и подходы в производство. Компания работает с широким спектром клиентов, предоставляя индивидуальные решения для упаковки, начиная от стандартных коробов и заканчивая сложными упаковочными конструкциями, такими как гофроподдоны и специализированные коробки.

Для обеспечения бесперебойной работы и высокого качества продукции, предприятие акцентирует внимание на оптимизации всех бизнес-процессов [9]. Одним из важнейших аспектов работы компании является эффективное управление складом, что напрямую влияет на своевременность поставок, соблюдение сроков и качество обслуживания клиентов.

Процесс адаптации и оптимизации складских операций требует непрерывного совершенствования. Используем метод структурного анализа для выявления ключевых блоков в процессе управления складом и их оптимизации [6].

Использую нотацию IDEF0, чтобы смоделировать и описать модель организации [5]. IDEF0 — это нотация, предназначенная для моделирования функциональных аспектов различных систем и процессов. Она используется для представления структуры и взаимодействий в сложных системах, таких

как бизнес-процессы, информационные системы, производственные процессы и другие.

Разработаю модель «как есть» в программе Ramus.

На рисунке 2 представлена контекстная диаграмма «Логистическая деятельность предприятия» вида «как есть».

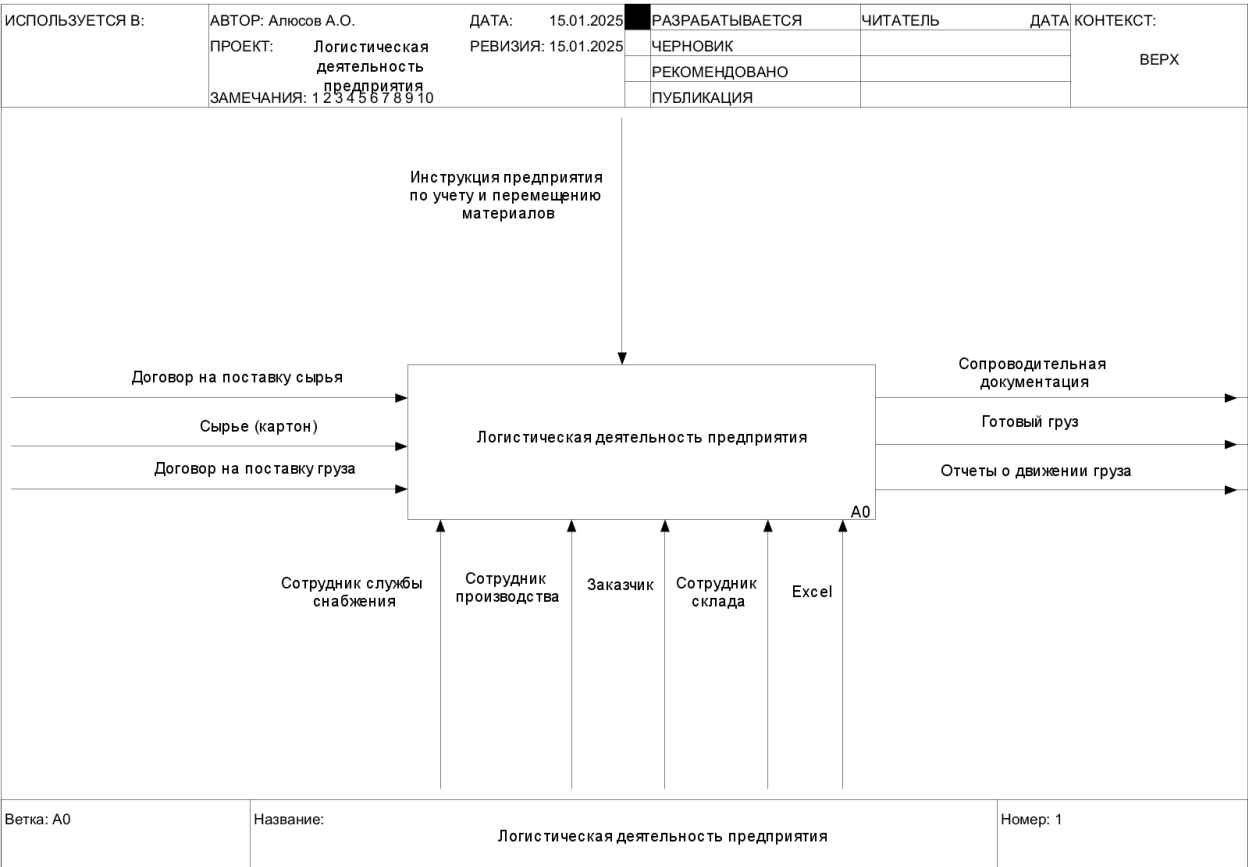


Рисунок 2 – Контекстная диаграмма «Логистическая деятельность предприятия» вида «как есть»

A0. Логистическая деятельность предприятия

На вход получаем договор на поставку сырья, сырье - картон, договор на поставку груза. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник службы снабжения, сотрудник производства, заказчик, сотрудник склада, Excel. На выход получаем сопроводительную документацию, готовый груз, отчеты о движении груза.

На рисунке 3 представлена декомпозиция контекстной диаграммы «Логистическая деятельность предприятия» вида «как есть».

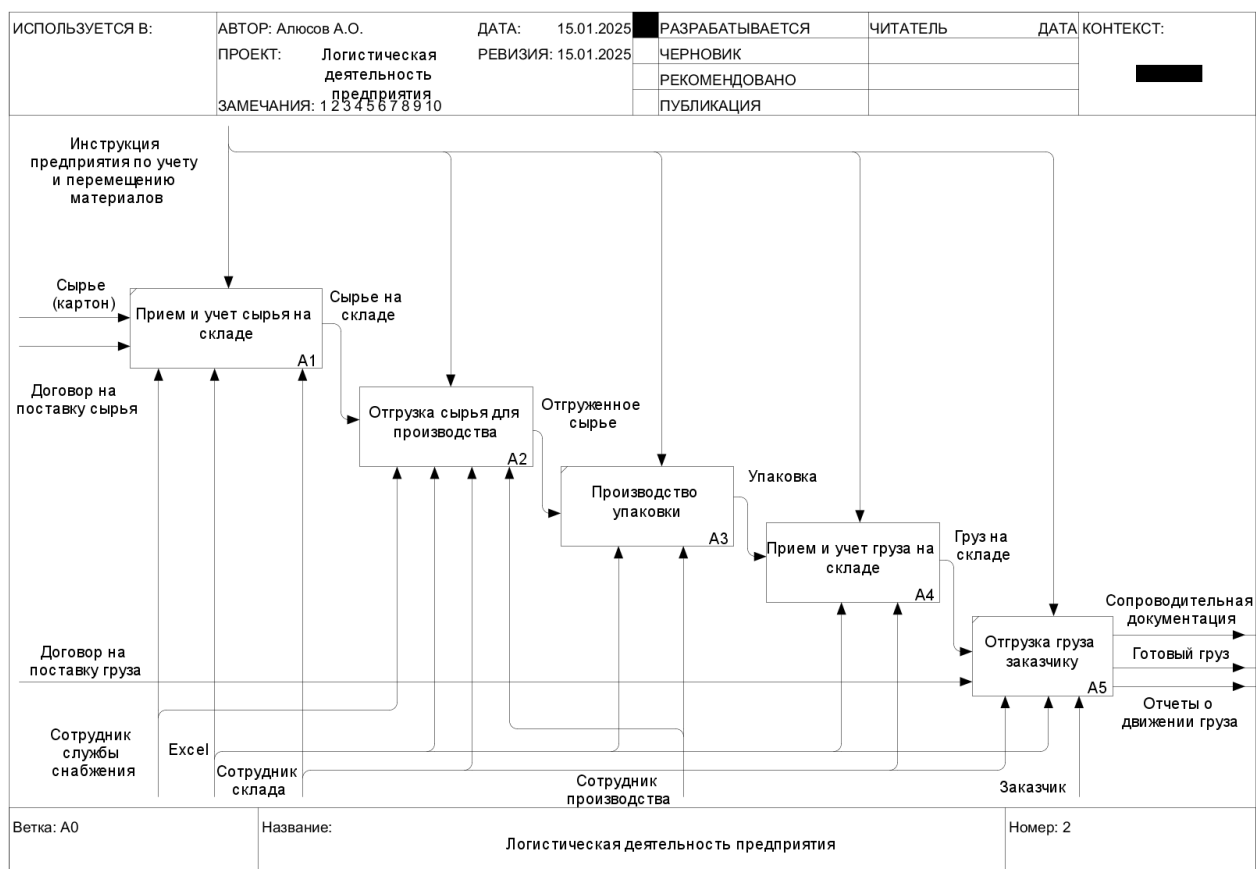


Рисунок 3 – Декомпозиция контекстной диаграммы «Логистическая деятельность предприятия» вида «как есть»

A1. Прием и учет сырья на складе

На вход получаем сырье - картон и договор на поставку сырья. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада, Excel и сотрудник службы снабжения. На выход получаем сырье на складе.

A2. Отгрузка сырья для производства

На вход получаем сырье на складе. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада, Excel, сотрудник службы

снабжения и сотрудник производства. На выход получаем отгруженное сырье.

А3. Производство упаковки

На вход получаем отгруженное сырье. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник производства и Excel. На выход получаем упаковку.

А4. Прием и учет груза на складе

На вход получаем упаковку. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада и Excel. На выход получаем груз на складе.

А5. Отгрузка груза заказчику

На вход получаем груз на складе и договор на поставку груза. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада, Excel и заказчик. На выход получаем сопроводительную документацию, готовый груз, отчеты о движении груза.

Диаграмма IDEF0 описывает последовательность процессов, связанных с логистикой склада и производством упаковки [1]. Каждый процесс опирается на входные данные: договоры, сырье, подчиняется определенным регламентам, выполняется сотрудниками склада с использованием Excel, и выдает выходные данные, соответствующие этапу процесса.

Представляю декомпозицию нескольких процессов. Декомпозиция процесса в модели IDEF0 — это один из ключевых этапов построения модели, направленный на детальное представление иерархии процессов. Она позволяет разложить сложный на уровни, где каждый уровень описывает действия с различной степенью детализации.

На рисунке 4 представлена диаграмма декомпозиции процесса «Отгрузка сырья для производства» вида «как есть».

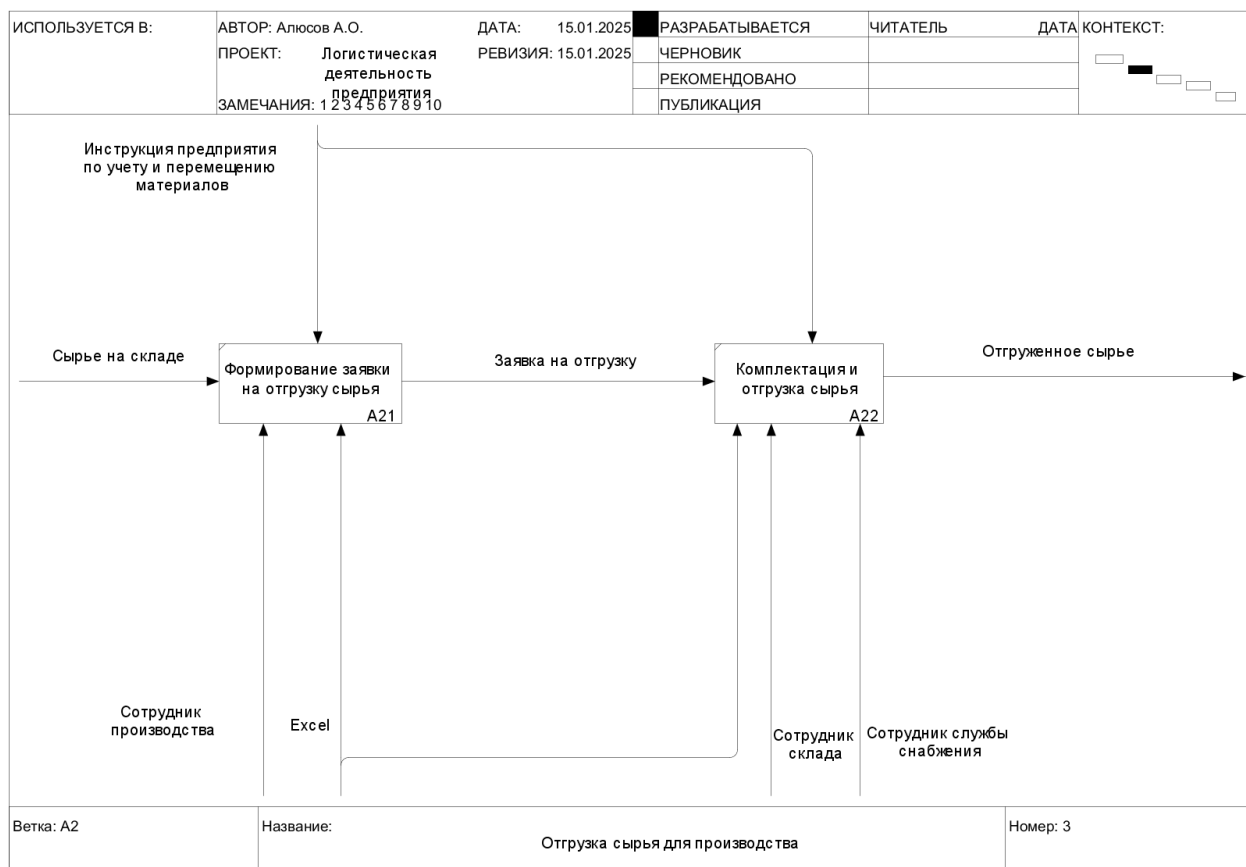


Рисунок 4 – Диаграмма декомпозиции процесса «Отгрузка сырья для производства» вида «как есть»

A21 Формирование заявки на отгрузку сырья

На вход получаем сырье на складе. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник производства и Excel. На выход получаем заявку на отгрузку.

A22 Комплектация и отгрузка сырья

На вход получаем заявку на отгрузку. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада, сотрудник службы снабжения и Excel. На выход получаем отгруженное сырье.

На рисунке 5 представлена диаграмма декомпозиции процесса «Прием и учет груза на складе» вида «как есть».

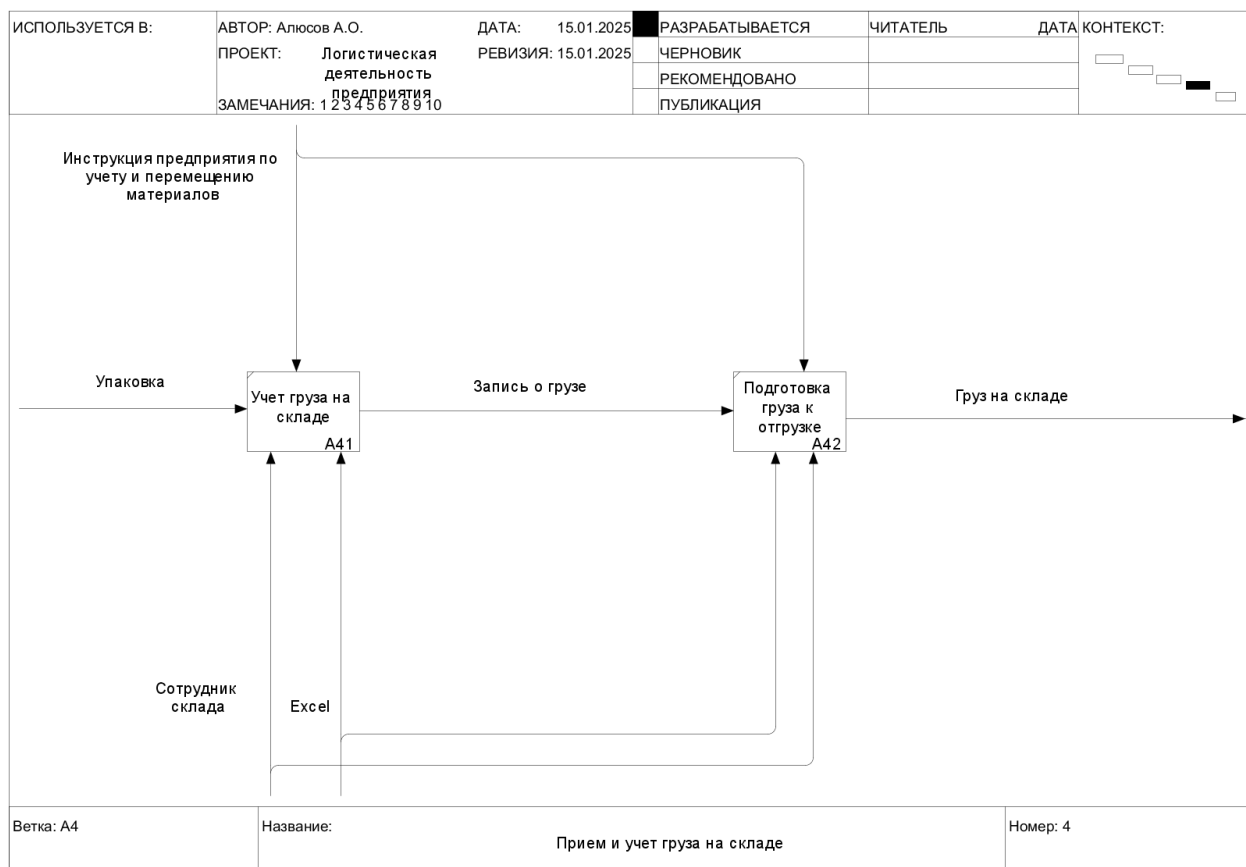


Рисунок 5 – Диаграмма декомпозиции процесса «Прием и учет груза на складе» вида «как есть»

A41 Учет груза на складе

На вход получаем упаковку. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада и Excel. На выход получаем запись о грузе.

A42 Подготовка груза к отгрузке

На вход получаем запись о грузе. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада и Excel. На выход получаем груз на складе.

1.4 Функциональная модель организации «как должно быть»

Компания «Волжский Картон» продолжает улучшать внутренние процессы для обеспечения еще более высокой скорости, точности и

надежности операций. Внедрение новых решений и инструментов, таких как веб-приложения становится важным шагом для автоматизации и оптимизации всех этапов работы. В частности, переход от использования Excel к специализированным веб-приложениям позволяет значительно повысить точность учета, ускорить процессы и обеспечить более удобный доступ к данным для всех участников процесса.

На рисунке 6 представлена контекстная диаграмма «Логистическая деятельность предприятия» вида «как должно быть».

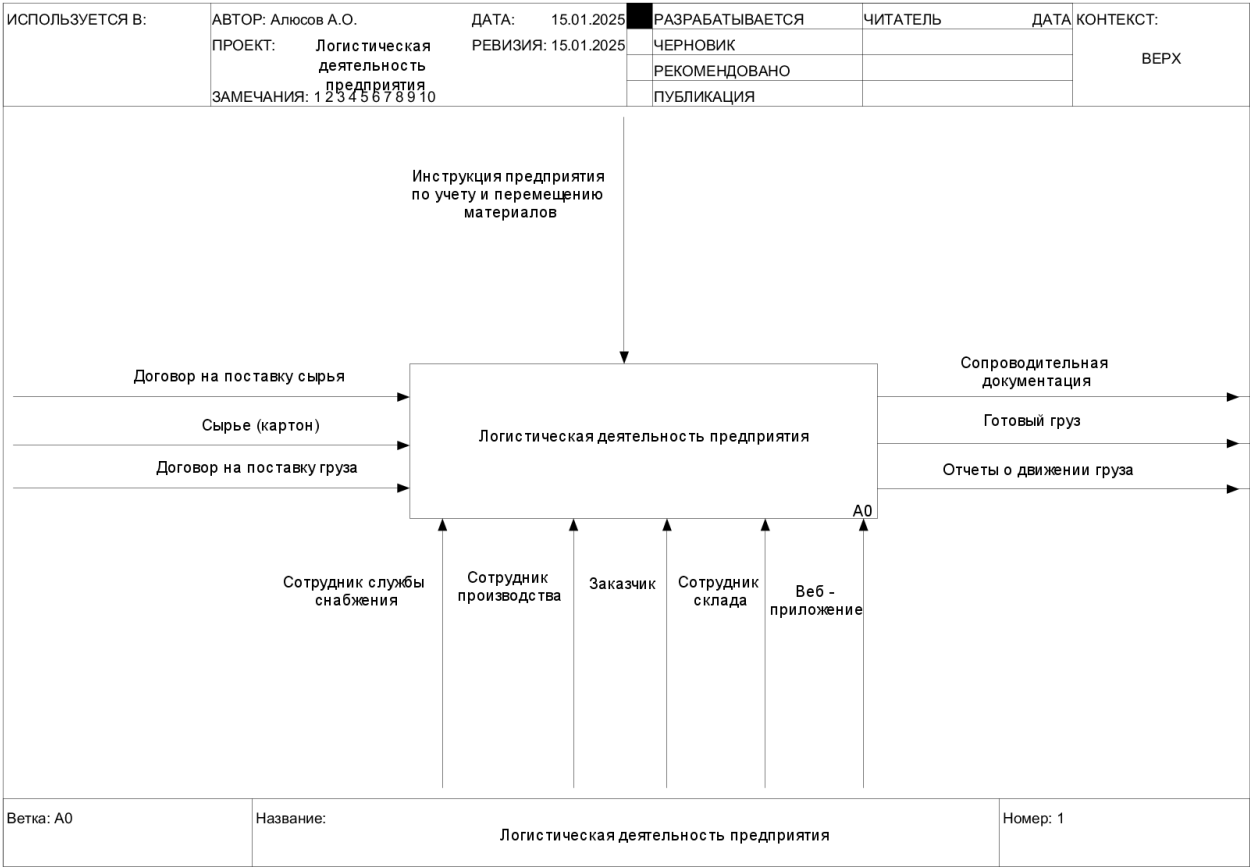


Рисунок 6 – Контекстная диаграмма «Логистическая деятельность предприятия» вида «как должно быть»

А0. Логистическая деятельность предприятия

На вход получаем договор на поставку сырья, сырье - картон, договор на поставку груза. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник

службы снабжения, сотрудник производства, заказчик, сотрудник склада, веб - приложение. На выход получаем сопроводительную документацию, готовый груз, отчеты о движении груза.

На рисунке 7 представлена декомпозиция контекстной диаграммы «Логистическая деятельность предприятия» вида «должно быть».

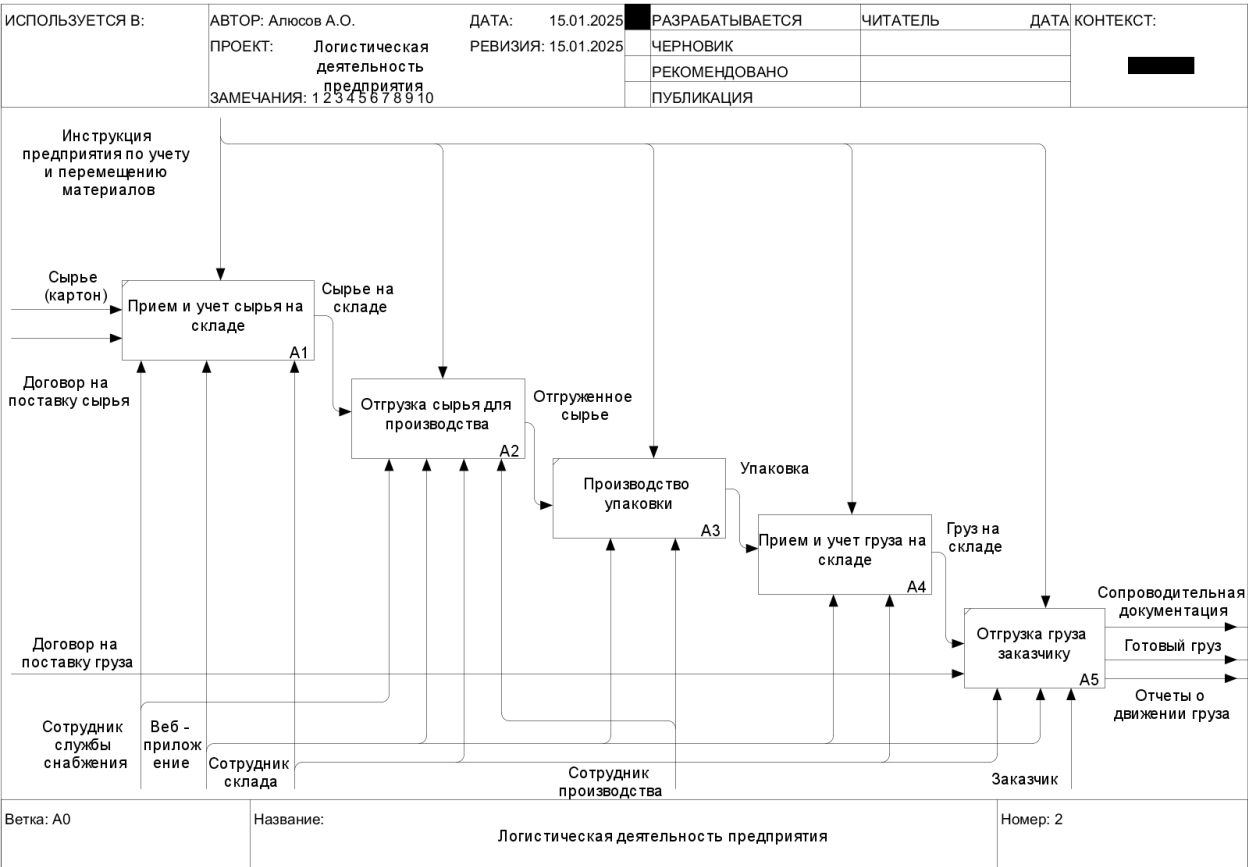


Рисунок 7 – Декомпозиция контекстной диаграммы «Логистическая деятельность предприятия» вида «как должно быть»

A1. Прием и учет сырья на складе

На вход получаем сырье - картон и договор на поставку сырья. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада, веб - приложение и сотрудник службы снабжения. На выход получаем сырье на складе.

A2. Отгрузка сырья для производства

На вход получаем сырье на складе. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада, веб - приложение, сотрудник службы снабжения и сотрудник производства. На выход получаем отгруженное сырье.

А3. Производство упаковки

На вход получаем отгруженное сырье. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник производства и веб - приложение. На выход получаем упаковку.

А4. Прием и учет груза на складе

На вход получаем упаковку. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада и веб - приложение. На выход получаем груз на складе.

А5. Отгрузка груза заказчику

На вход получаем груз на складе и договор на поставку груза. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада, веб - приложение и заказчик. На выход получаем сопроводительную документацию, готовый груз, отчеты о движении груза.

Модель с использованием веб-приложения представляет более современный подход к управлению логистикой на складе. Она упрощает работу сотрудников, минимизирует ошибки и улучшает управляемость процессов благодаря автоматизации.

На рисунке 8 представлена диаграмма декомпозиции процесса «Отгрузка сырья для производства» вида «как должно быть».

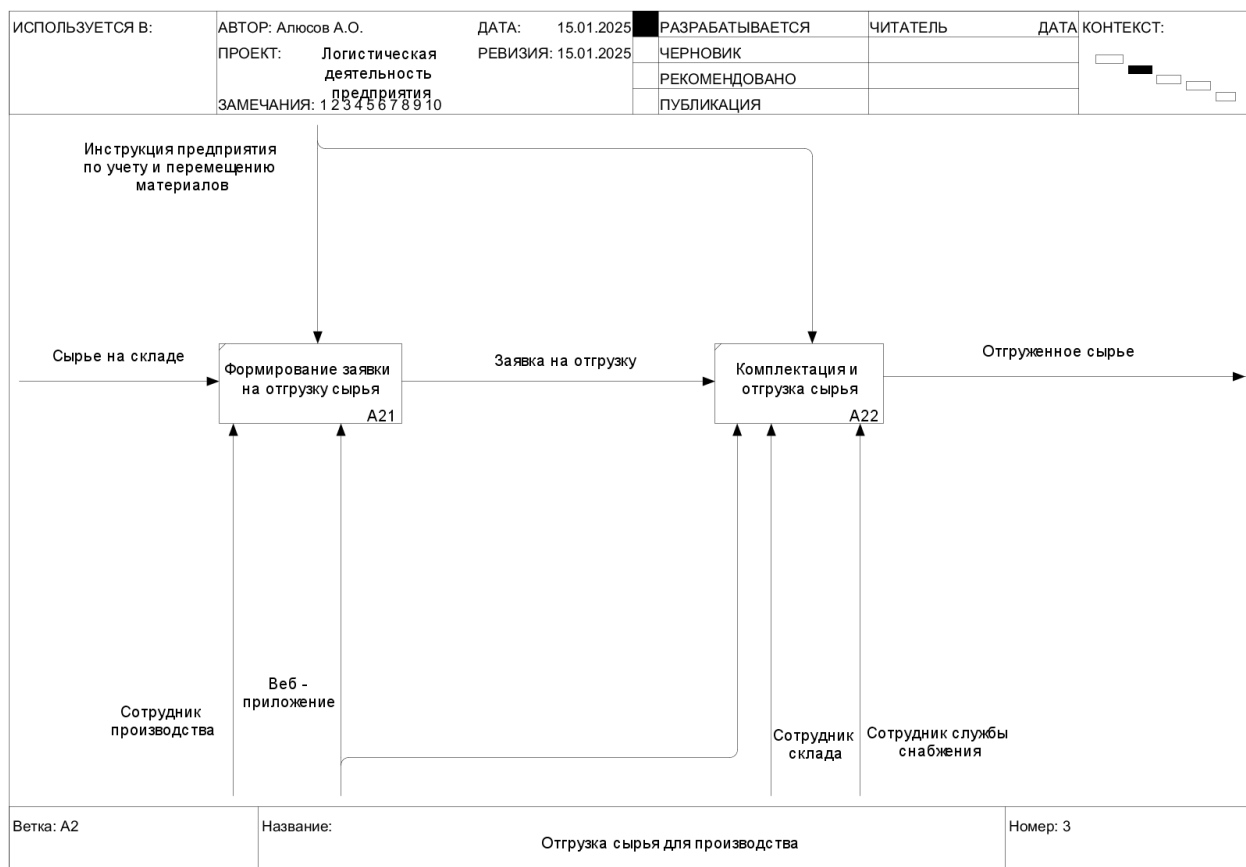


Рисунок 8 – Диаграмма декомпозиции процесса «Отгрузка сырья для производства» вида «как должно быть»

A21 Формирование заявки на отгрузку сырья

На вход получаем сырье на складе. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник производства и веб - приложение. На выход получаем заявку на отгрузку.

A22 Комплектация и отгрузка сырья

На вход получаем заявку на отгрузку. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада, сотрудник службы снабжения и веб - приложение. На выход получаем отгруженное сырье.

На рисунке 9 представлена диаграмма декомпозиции процесса «Прием и учет груза на складе» вида «как должно быть».

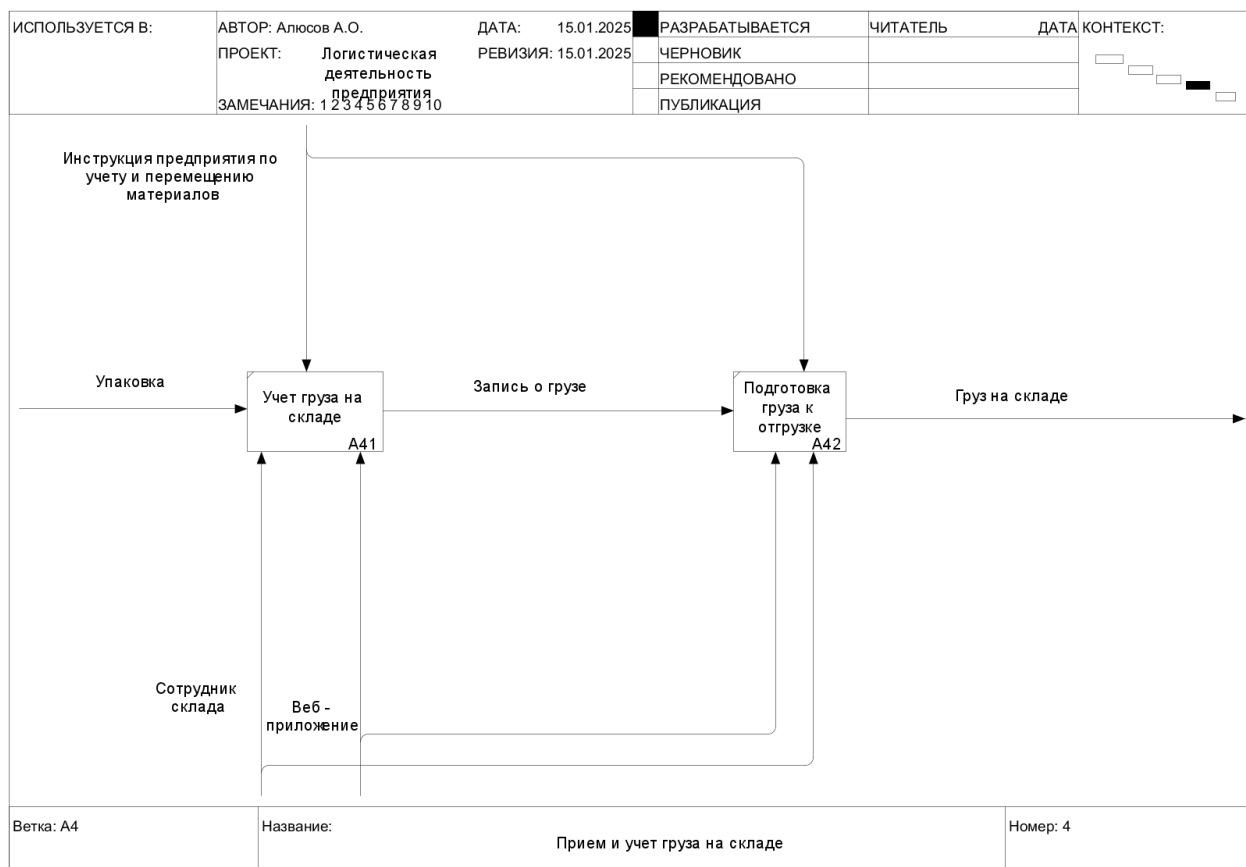


Рисунок 9 – Диаграмма декомпозиции процесса «Прием и учет груза на складе» вида «как должно быть»

A41 Учет груза на складе

На вход получаем упаковку. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада и веб - приложение. На выход получаем запись о грузе.

A42 Подготовка груза к отгрузке

На вход получаем запись о грузе. Управляется этот блок инструкциями предприятия по учету и перемещению материалов. К механизмам относятся сотрудник склада и веб - приложение. На выход получаем груз на складе.

Диаграмма IDEF0 показывает, что логистическая деятельность предприятия четко разделена на этапы, где веб-приложение играет ключевую роль в учете, оформлении заявок и контроле движения грузов. Автоматизация процессов необходима для повышения точности, скорости и согласованности операций.

1.5 Анализ лучших практик и существующих разработок в предметной области

Складской учет и управление запасами — ключевая часть логистической и операционной деятельности любой компании. Эффективная работа склада способствует оптимизации процессов, снижению затрат, улучшению обслуживания клиентов и повышению прибыли. В последние годы на смену традиционным методам учета и обработки грузов приходят высокотехнологичные решения, которые значительно упрощают и автоматизируют эти процессы.

Современные системы управления складом предлагают разнообразные функции, такие как автоматизация учета груза, интеграция с внешними системами, использование мобильных устройств для инвентаризаций, автоматическое создание отчетности и прогнозирование спроса. Все эти инструменты помогают повысить точность учета, ускорить обработку и снизить вероятность ошибок.

Проанализирую лучшие практики, на примерах конкретных программных обеспечений.

SAP Extended Warehouse Management (EWM) — это мощная система управления складом, интегрируемая с ERP-системами SAP, и она наиболее подходит для крупных предприятий, которым требуется автоматизация учета и высокая степень интеграции. SAP EWM поддерживает использование мобильных устройств, таких как сканеры штрих-кодов, а также автоматическое создание отчетности, что упрощает учет движения груза, управление запасами и прогнозирование. Решение позволяет оптимизировать процессы на основе данных, предоставляя продвинутую аналитику. SAP EWM довольно дорогое и сложное в настройке ПО, что делает его избыточным для небольших складов с простыми операциями, но оно идеально подходит для больших логистических центров, где необходима полная автоматизация и синхронизация со всей цепочкой поставок.

Odoo Inventory — это гибкое и доступное решение для управления складом, которое хорошо подходит для малого и среднего бизнеса. Odoo предлагает модульный подход, позволяющий компаниям добавлять и настраивать необходимые функции, включая управление запасами, интеграцию с CRM и ERP, а также автоматизацию отчетности. Эта система поддерживает подключение мобильных устройств, сканеров штрих-кодов и обеспечивает управление запасами с прогнозированием спроса. Она эффективна для автоматизации учета и управления запасами, а также для создания отчетности. Учитывая доступную стоимость и гибкость, Odoo может хорошо подойти для складов среднего размера.

Microsoft Dynamics 365 Supply Chain Management (SCM). Эта система включает инструменты для складского учета, оптимизации запасов, интеграции с ERP и CRM, а также поддержку мобильных устройств и сканеров штрих-кодов. Несмотря на высокую стоимость, Microsoft Dynamics 365 SCM является эффективным решением для крупных предприятий, которым требуется комплексное управление логистическими процессами.

Fishbowl представляет собой складскую систему, ориентированную на средний бизнес. Её ключевым преимуществом является интеграция с QuickBooks, что упрощает ведение учета и финансов. Fishbowl предоставляет базовые инструменты аналитики, автоматизированную отчетность и возможность использования сканеров штрих-кодов. Однако функциональные ограничения делают её менее подходящей для сложных логистических процессов.

NetSuite Warehouse Management System (WMS) — мощное решение для автоматизации складского учета, поддерживающее мобильные устройства и интеграцию с ERP-системами. Эта система предлагает расширенные аналитические возможности, что делает её привлекательной для крупных предприятий. Однако высокая стоимость может стать препятствием для небольших компаний.

QuickBooks Commerce предназначен для автоматизации учета и управления запасами, а также интегрируется с бухгалтерским ПО QuickBooks. Эта система удобна для малых складов, но имеет ограниченные возможности масштабирования и не подходит для крупных предприятий с комплексными логистическими процессами.

Анализ существующих решений показал, что ни одна из представленных систем не удовлетворяет в полной мере специфическим требованиям ООО «Волжский картон». В связи с этим было принято решение о разработке специализированного веб-приложения, которое позволит повысить точность и прозрачность учета, упростить работу с отчетностью и анализом данных.

1.6 Функциональные требования проекта

Для определения требований к разрабатываемому приложению используется методологию FURPS+.

Методология FURPS+ - это модель классификации требований к программному обеспечению [15]. Она охватывает функциональность, удобство использования, надежность, производительность, поддерживаемость и дополнительные аспекты.

В таблице 1 представлены основные требования к разрабатываемому приложению с учетом методологии FURPS+.

Таблица 1 – Классификация требований к разрабатываемому приложению

Требование	Статус	Полезность	Риск
Функциональные требования			
Отображение груза на складе	Важный	Высокая	Средний
Добавление нового груза на склад	Важный	Высокая	Средний

Продолжение таблицы 1

Требование	Статус	Полезность	Риск
Функциональные требования			
Редактирование записи о грузе	Важный	Средняя	Средний
Удаление записи о грузе	Важный	Высокая	Средний
Фильтрация груза по названию	Важный	Средняя	Средний
Отгрузка груза	Важный	Высокая	Высокий
Просмотр истории отгрузок	Важный	Средняя	Средний
Удобство использования			
Интуитивно понятный интерфейс	Важное	Высокий	Высокий
Наличие формы для добавления груза	Важное	Высокий	Средний
Адаптивный дизайн	Важное	Высокий	Средний
Возможность поиска и фильтрации груза	Важное	Высокий	Средний
Легкость редактирования записи о грузе	Важное	Высокий	Средний
Надежность			
Защита данных	Важный	Высокий	Высокий
Высокая доступность системы	Важный	Высокий	Высокий
Производительность			
Быстрое выполнение операций	Важный	Высокий	Средний
Обработка большого объема данных	Важный	Средний	Высокий
Легкость в обновлении	Важный	Высокий	Средний
Возможность интеграции с другими системами	Важный	Высокий	Средний
Ограничения			
Ограничения по времени внедрения	Важный	Высокая	Высокий
Бюджетные ограничения	Важный	Высокая	Высокий

При разработке веб – приложения нужно учитывать вышеописанные требования.

Выводы по главе 1

В первой главе выпускной квалификационной работы проанализирована деятельность ООО «Волжский картон». Рассмотрены цели, задачи и особенности работы организации, а также ее структура и распределение обязанностей между сотрудниками.

В ходе исследования была построена модель текущих бизнес-процессов предприятия в нотации IDEF0, а также предложены возможные улучшения. На основе проведенного анализа сформулированы основные задачи и цели внедрения информационной системы.

Особое внимание было уделено требованиям к разработке информационной системы для автоматизации деятельности склада. При формулировании требований использована методология FURPS. Были выделены критически важные функции, такие как отображение груза на складе и возможность добавления нового груза. Также акцент сделан на интуитивно понятный интерфейс, быструю обработку операций и надёжную защиту данных.

Дополнительно был проведен обзор лучших решений в предметной области, что позволило обосновать выбор подхода к автоматизации.

Глава 2 Архитектура проекта и особенности его реализации

2.1 Диаграмма прецедентов веб - приложения

Для разработки концептуальной модели проекта применяется унифицированный язык моделирования – UML [11]. В частности, используется диаграмма вариантов использования - Use Case Diagram, которая наглядно демонстрирует функциональные возможности создаваемой системы и их распределение между различными категориями пользователей.

Создание диаграммы выполняется с помощью программного инструмента StarUML.

На рисунке 10 представлена диаграмма вариантов использования для разрабатываемого веб-приложения.

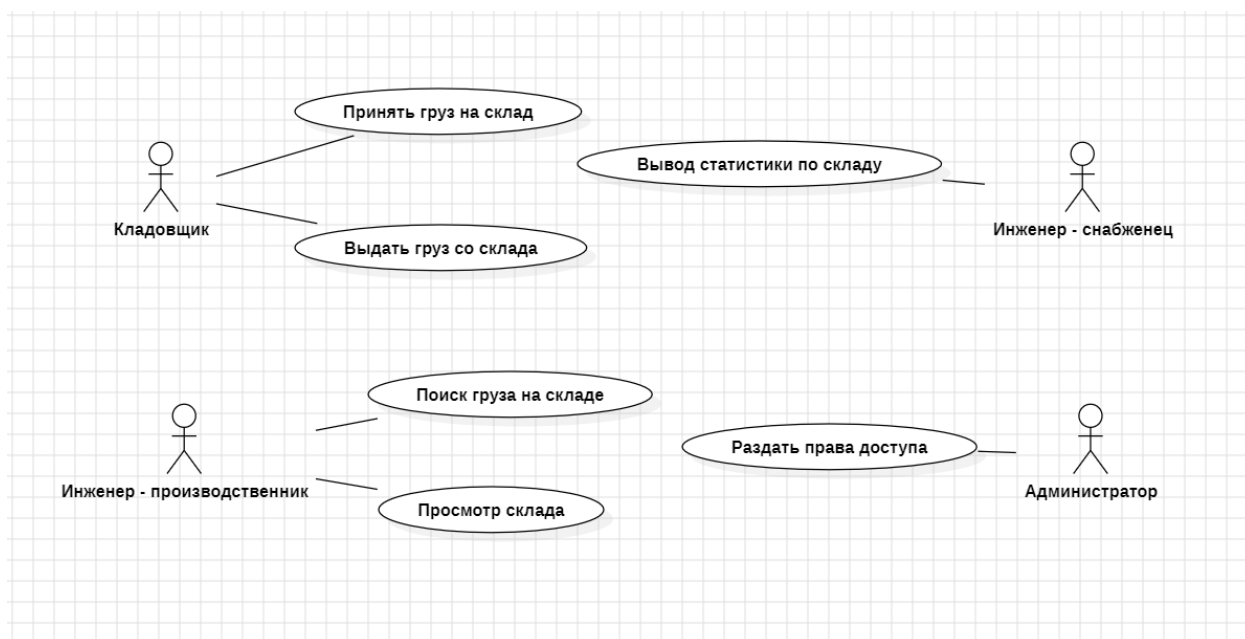


Рисунок 10 – Диаграмма прецедентов

Диаграмма вариантов использования демонстрирует основные функции веб-приложения, распределенные между разными категориями пользователей.

В таблицах 2 – 7 представлены прецеденты.

Таблица 2 – Выдать груз со склада

Поле	Значение
Актор	Кладовщик
Цель	Списание груза, покидающего склад
Входные данные	Информация о грузе: идентификатор, количество
Выходные данные	Обновленные данные о запасах на складе
Сценарий	1. Кладовщик выбирает груз для отпуска
	2. Указывает количество и подтверждает отпуск
	3. Система обновляет остатки на складе
Альтернативы	Если запрашиваемого количества нет, выдается сообщение о недостатке запасов

Таблица 3 – Принять груз на склад

Поле	Значение
Актор	Кладовщик
Цель	Прием груза на склад
Входные данные	Информация о грузе: идентификатор, количество, тип и т.д.
Выходные данные	Обновленные данные о запасах на складе
Сценарий	1. Кладовщик регистрирует поступление груза
	2. Указывает детали: количество, тип и т.д.
	3. Система обновляет текущие запасы
Альтернативы	Если превышена вместимость склада, система предупреждает об этом

Таблица 4 – Просмотр склада

Поле	Значение
Актор	Инженер – производитель
Цель	Получение общей информации о складе
Входные данные	Запрос на просмотр текущего состояния склада
Выходные данные	Данные о текущих запасах
Сценарий	1. Инженер отправляет запрос на просмотр склада
	2. Система возвращает текущие данные

Таблица 5 – Поиск груза на складе

Поле	Значение
Актор	Инженер – производитель
Цель	Поиск конкретного груза на складе
Входные данные	Параметры поиска: название, тип, идентификатор
Выходные данные	Информация о найденном грузе
Сценарий	1. Инженер указывает параметры поиска
	2. Система возвращает результаты поиска
Альтернативы	Если груз не найден, система уведомляет об этом

Таблица 6 – Вывод статистики по складу

Поле	Значение
Актор	Инженер – снабженец
Цель	Получение статистики для анализа запасов
Входные данные	Запрос на статистику по складу
Выходные данные	Статистические данные: остатки, динамика использования
Сценарий	1. Инженер отправляет запрос на статистику
	2. Система генерирует и возвращает статистику

Таблица 7 – Раздать права доступа

Поле	Значение
Актор	Администратор
Цель	Управление доступом пользователей
Входные данные	Информация о пользователе и его роли
Выходные данные	Обновленные данные о правах доступа
Сценарий	1. Администратор выбирает пользователя
	2. Назначает или изменяет права доступа
	3. Система сохраняет изменения
Альтернативы	Если пользователь не найден, процесс отклоняется

Этот перечень вариантов использования четко определяет границы функциональности системы, что способствует разграничению ролей пользователей и оптимальному распределению задач внутри команды.

Благодаря такому подходу каждый участник процесса имеет доступ только к необходимым ему операциям, что повышает эффективность работы и безопасность данных.

2.2 Диаграмма классов разрабатываемого веб - приложения

Диаграмма классов является одним из ключевых инструментов UML, позволяющим наглядно представить структуру системы, включая классы, их атрибуты, методы и связи между ними [17].

При разработке веб-приложений можно выделить несколько основных сущностей, каждая из которых выполняет определенную функцию и взаимодействует с другими элементами системы. Такое распределение ролей обеспечивает слаженную работу всего приложения.

На рисунке 11 представлена диаграмма классов, демонстрирующая архитектурное устройство веб-приложения.

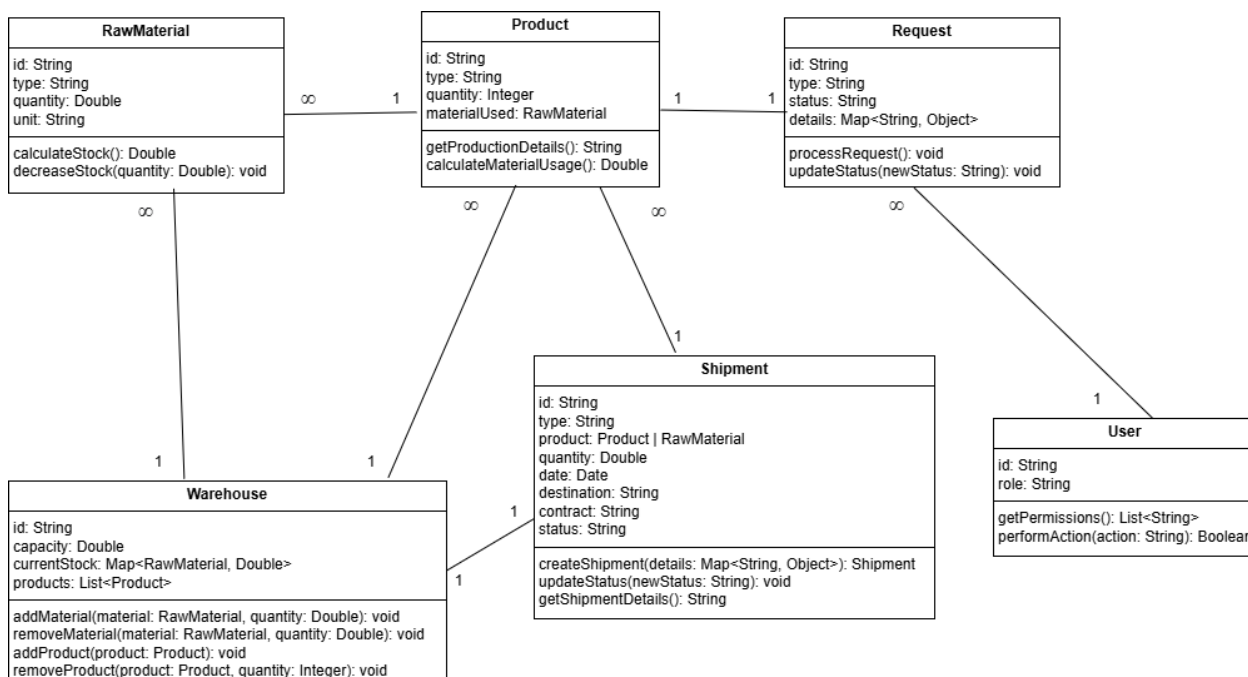


Рисунок 11 – Диаграмма классов

В таблицах 8 – 13 представлена диаграмма классов в табличной форме

Таблица 8 – Класс RawMaterial

Атрибут / Метод	Тип	Описание
id	String	Уникальный идентификатор сырья
type	String	Тип картона, например, гофрокартон
quantity	Double	Количество сырья
unit	String	Единица измерения, например, квадратные метры
calculateStock()	Double	Метод для расчета доступного остатка
decreaseStock(quantity)	void	Метод для уменьшения запаса сырья

Таблица 9 – Класс Product

Атрибут / Метод	Тип	Описание
id	String	Уникальный идентификатор продукции
type	String	Тип упаковки, например, коробка
quantity	Integer	Количество готовой продукции
materialUsed	RawMaterial	Сырье, использованное для создания упаковки
getProductionDetails()	String	Метод для получения информации о производстве упаковки
calculateMaterialUsage()	Double	Метод для расчета использованного сырья

Таблица 10 – Класс Warehouse

Атрибут / Метод	Тип	Описание
id	String	Уникальный идентификатор склада
capacity	Double	Общая вместимость склада
currentStock	Map<RawMaterial, Double>	Текущий запас сырья
products	List<Product>	Список готовой продукции

Продолжение таблицы 10

Атрибут / Метод	Тип	Описание
addMaterial(material, quantity)	void	Метод для добавления сырья на склад
removeMaterial(material, quantity)	void	Метод для списания сырья со склада
addProduct(product)	void	Метод для добавления готовой продукции
removeProduct(product, quantity)	void	Метод для списания готовой продукции

Таблица 11 – Класс Request

Атрибут / Метод	Тип	Описание
id	String	Уникальный идентификатор заявки
type	String	Тип заявки
status	String	Статус заявки
details	Map<String, Object>	Информация о заявке
processRequest()	void	Метод для обработки заявки
updateStatus(newStatus)	void	Метод для обновления статуса заявки

Таблица 12 – Класс User

Атрибут / Метод	Тип	Описание
id	String	Уникальный идентификатор пользователя
role	String	Роль пользователя
getPermissions()	List<String>	Метод для получения списка прав доступа
performAction(action)	Boolean	Метод для выполнения действия в зависимости от роли

Таблица 13 – Класс Shipment

Атрибут / Метод	Тип	Описание
id	String	Уникальный идентификатор отгрузки

Продолжение таблицы 13

Атрибут / Метод	Тип	Описание
Type	String	Тип отгрузки
product	Product RawMaterial	Продукция или сырье, связанное с отгрузкой
Quantity	Double	Количество отгружаемого груза
date	date	Дата отгрузки
destination	String	Место назначения
contract	String	Связанный договор
status	String	Статус отгрузки
createShipment(details)	Shipment	Метод для создания новой отгрузки
updateStatus(newStatus)	void	Метод для обновления статуса отгрузки
getShipmentDetails()	String	Метод для получения информации об отгрузке

Диаграмма классов подробно отразила внутреннюю структуру системы, включая классы, их свойства, методы и взаимосвязи [2]. Этот инструмент моделирования сыграл значительную роль в проектировании архитектуры, позволив наглядно представить ключевые компоненты, определить их назначение и уточнить взаимодействие между объектами. Благодаря диаграмме была сформирована логически организованная структура, в которой каждый класс выполняет строго определенную задачу, что способствует модульности системы и упрощает ее дальнейшую поддержку.

2.3 Логическая модель данных веб - приложения

Логическая модель данных в UML представляет собой графическое отображение структуры данных, используемое для описания объектов, их атрибутов и взаимосвязей в рамках конкретного приложения [21]. Этот инструмент играет важную роль в проектировании информационных систем, позволяя наглядно представить организацию данных и установить логические связи между элементами еще до их реализации в базе данных.

На основе ранее разработанных классов и установленных связей между ними, логическая модель данных будет включать следующие элементы:

- сущности — основные объекты, которые характеризуют элементы предметной области. Каждая сущность представляет собой отдельную таблицу или объект в будущей базе данных.
- атрибуты — свойства или характеристики сущностей, определяющие их уникальные особенности и поведение. Атрибуты могут быть обязательными или необязательными, а также иметь типы данных, соответствующие их значениям (например, строка, число, дата).
- связи между сущностями — логические отношения, описывающие, как объекты взаимодействуют друг с другом.

На основе описанных ранее классов и их связей логическая модель данных будет включать сущности и их атрибуты, а также связи между этими сущностями (рисунок 12):



Рисунок 12 – Логическая модель данных

Следующим шагом создам физическую модель данных [3]. Физическая модель данных — это одна из ключевых стадий проектирования баз данных,

на которой определяется, как данные будут храниться и организовываться в реальной системе хранения [8]. Она представляет собой детализированную структуру данных, включая спецификацию таблиц, индексов, связей, типов данных и других характеристик, необходимых для эффективного хранения и доступа к данным в конкретной среде исполнения.

Правильное проектирование физической модели позволяет не только повысить производительность и масштабируемость системы, но и обеспечить её стабильность, отказоустойчивость и безопасность.

На рисунке 13 показана физическая модель данных.

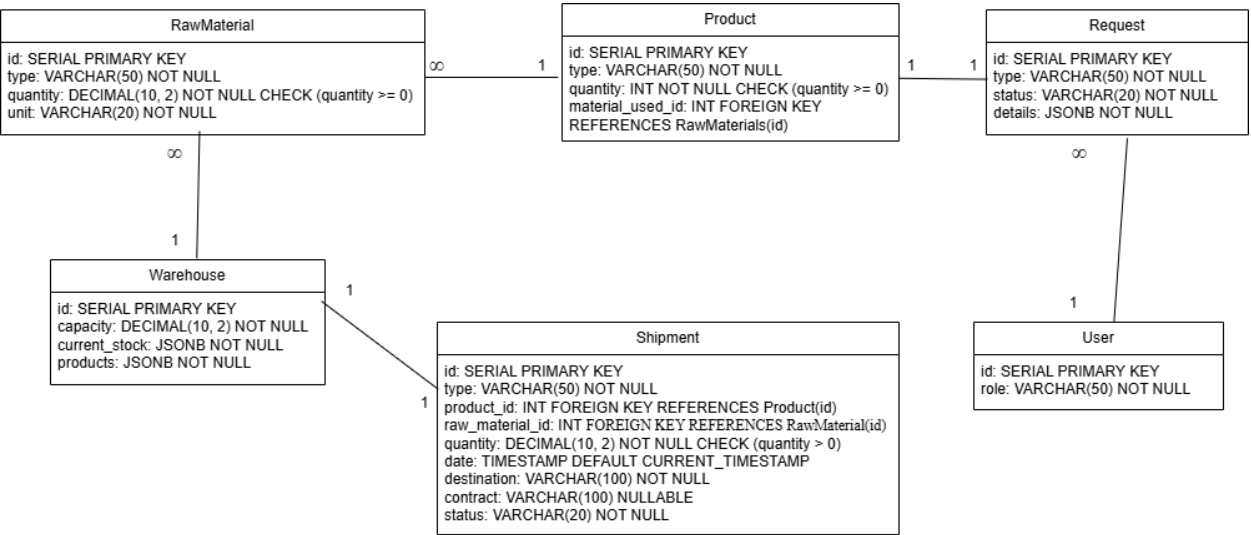


Рисунок 13 – Физическая модель данных

После создания логической и физической модели данных я представлю их соответствие.

В таблице 14 представлены соответствие сущностей логической и физической моделей.

Таблица 14 – Соответствие сущностей

Сущность	Таблица
----------	---------

Продолжение таблицы 14

Сущность	Таблица
Сырье	RawMaterial
Упаковка	Product
Склад	Warehouse
Заявки	Request
Пользователи	User
Отгрузки	Shipment

После создания физической модели данных, а также приведения соответствий сущностей представлю её в виде таблиц.

В таблицах 15 – 20 представлена физическая модель данных в табличной форме.

Таблица 15 – Таблица RawMaterial

Атрибут	Тип данных	Описание	Ограничения
id	SERIAL	Уникальный идентификатор сырья	PRIMARY KEY
type	VARCHAR(50)	Тип картона	NOT NULL
quantity	DECIMAL(10, 2)	Количество сырья	NOT NULL, CHECK (quantity >= 0)
unit	VARCHAR(20)	Единица измерения	NOT NULL

Таблица 16 – Таблица Product

Атрибут	Тип данных	Описание	Ограничения
id	SERIAL	Уникальный идентификатор продукции	PRIMARY KEY
type	VARCHAR(50)	Тип упаковки	NOT NULL
quantity	INT	Количество продукции	NOT NULL, CHECK (quantity >= 0)
material_used_id	INT	Ссылка на использованное сырье	FOREIGN KEY REFERENCES RawMaterials(id)

Таблица 17 – Таблица Warehouse

Атрибут	Тип данных	Описание	Ограничения
Id	SERIAL	Уникальный идентификатор склада	PRIMARY KEY
capacity	DECIMAL(10, 2)	Общая вместимость склада	NOT NULL
current_stock	JSONB	Текущий запас сырья в формате JSON	NOT NULL
products	JSONB	Запасы готовой продукции в формате JSON	NOT NULL

Таблица 18 – Таблица Request

Атрибут	Тип данных	Описание	Ограничения
id	SERIAL	Уникальный идентификатор заявки	PRIMARY KEY
type	VARCHAR(50)	Тип заявки	NOT NULL
status	VARCHAR(20)	Статус заявки	NOT NULL
details	JSONB	Информация о заявке	NOT NULL

Таблица 19 – Таблица User

Атрибут	Тип данных	Описание	Ограничения
id	SERIAL	Уникальный идентификатор пользователя	PRIMARY KEY
role	VARCHAR(50)	Роль пользователя	NOT NULL

Таблица 20 – Таблица Shipment

Атрибут	Тип данных	Описание	Ограничения
id	SERIAL	Уникальный идентификатор отгрузки	PRIMARY KEY
type	VARCHAR(50)	Тип отгрузки	NOT NULL

Продолжение таблицы 20

Атрибут	Тип данных	Описание	Ограничения
product_id	INT	Ссылка на продукт	FOREIGN KEY REFERENCES Product(id)
raw_material_id	INT	Ссылка на сырье	FOREIGN KEY REFERENCES RawMaterial(id)
quantity	DECIMAL(10, 2)	Количество	NOT NULL, CHECK (quantity >= 0)
date	TIMESTAMP	Дата отгрузки	DEFAULT CURRENT_TIMESTAMP
destination	VARCHAR(100)	Место назначения	NOT NULL
contract	VARCHAR(100)	Договор	NULLABLE
status	VARCHAR(20)	Статус отгрузки	NOT NULL

Модель учитывает в себе отношения между элементами, отражает типы данных, первичные и внешние ключи. Построена с учетом СУБД PostgreSQL [13].

Выводы по главе 2

В результате проектирования системы был разработан архитектурный подход с разделением обязанностей между компонентами и выстраиванием моделей данных.

Приложение использует объектно-ориентированный подход, что упрощает поддержку, расширение и тестирование системы.

Логическая и физическая модели данных учитывают все сущности, атрибуты и связи, обеспечивая эффективное хранение и обработку данных, а также поддержание их целостности.

В физической модели данных предусмотрены индексы и оптимизация хранения для высокой производительности при больших объемах информации.

Диаграмма прецедентов и диаграмма классов обеспечивают интуитивно понятную систему взаимодействия с пользователями.

Глава 3 Физическое проектирование веб – приложения

3.1 Системная архитектура проекта

Приложение построено на основе клиент-серверной архитектуры, что обеспечивает четкое разделение обязанностей между серверной и клиентской частями, а также масштабируемость и гибкость системы [7]. Данная архитектура позволяет выделить ключевые компоненты, которые работают независимо друг от друга, обеспечивая легкость внесения изменений в приложение и упрощая процесс сопровождения [22]. Такое разделение также способствует улучшению безопасности, поскольку серверная часть изолирована от клиентской, а данные пользователей защищены на стороне сервера.

Серверная часть приложения реализована с использованием Express.js — популярного, легковесного и мощного фреймворка для создания веб-приложений [18]. Express.js обеспечивает богатый функционал для разработки серверной логики и поддерживать современные стандарты веб-технологий. В данном приложении серверная часть играет ключевую роль, обрабатывая запросы от клиента, выполняя вычисления и операции, а также взаимодействуя с базой данных PostgreSQL. Реализация всех операций CRUD (создание, чтение, обновление, удаление) строго структурирована, каждая операция имеет четко определенные маршруты и обработчики. Этот подход способствует модульности кода, упрощает отладку и дальнейшую модернизацию системы [10].

Для взаимодействия с базой данных PostgreSQL используется модуль pg, который является надежным инструментом для выполнения SQL-запросов и управления базой данных [19]. Он предоставляет удобные методы, такие как работа с подготовленными запросами, поддержка транзакций и управление пулом соединений. Это обеспечивает не только высокую производительность системы, но и позволяет эффективно управлять

ресурсами сервера. Поддержка подготовленных запросов защищает от SQL-инъекций, что повышает уровень безопасности приложения. Транзакции дают возможность гарантировать целостность данных, что особенно важно в системах, работающих с критически важной информацией.

Благодаря использованию современных технологий и инструментов, таких как Express.js и PostgreSQL, приложение обладает высокой надежностью, производительностью и гибкостью. Оно адаптировано для работы в условиях реального времени и способно масштабироваться.

В качестве демонстрации приведу пример программного кода инициализирующее серверное приложение на базе фреймворка Express и его настройке для работы с базой данных PostgreSQL (рисунок 14):

```
const express :e | () => core.Express = require('express');
const bodyParser :{...} | {...} = require('body-parser');
const { Pool } = require('pg');

const app :any | Express = express();
const pool = new Pool({
  user: 'postgres',
  host: 'localhost',
  database: 'program',
  password: '1806',
  port: 5432,
});

app.use(bodyParser.json());
app.use(express.static( root: 'public'));
```

Рисунок 14 – Инициализация и настройка приложения

Клиентская часть представляет собой интерактивное веб-приложение, разработанное с использованием HTML, CSS и JavaScript [4]. Она обеспечивает удобный интерфейс для взаимодействия пользователей с сервером. Благодаря использованию современных технологий, таких как Fetch API, приложение может отправлять запросы к серверу и обновлять данные на странице без необходимости перезагрузки [20].

Особое внимание при проектировании веб-приложения уделено дизайну пользовательского интерфейса – UI [16].

3.2 Описание функциональности веб – приложения

В рамках практической реализации я хотел бы остановиться на модуле для складского учета. Остальные модули приложения работают аналогичным образом.

Пользователь может использовать приложение через веб-интерфейс, который предоставляет доступ к функциям управления складом и отгрузками.

На главной странице отображаются секции для поиска и добавления, таблицы со списком груза на складе и историей отгрузок (рисунок 15):

Система складского учета и логистики

Поиск

Название Артикул SKU Количество Цена Масса в гр Длина в см
Ширина в см Высота в см **Добавить**

Склад

Название	Артикул SKU	Количество	Цена	Масса	Длина	Ширина	Высота	Дата поступления	Действия
----------	-------------	------------	------	-------	-------	--------	--------	------------------	----------

История отгрузок

Груз	Количество	Артикул SKU	Дата отгрузки
------	------------	-------------	---------------

Рисунок 15 – Веб – интерфейс

Для добавления, например, гофрокороба на склад кладовщику необходимо будет заполнить соответствующие поля и нажать на кнопку добавить, после чего груз отобразится на складе (рисунок 16):

Система складского учета и логистики

Поиск

Гофрокороб

R987

20

150

340

30

340

14,2

Добавить

Склад

Название	Артикул SKU	Количество	Цена	Масса	Длина	Ширина	Высота	Дата поступления	Действия
Гофрокороб	R987	20	150.00	340.00	30.00	340.00	14.20	06.01.2025, 23:03:45	Редактировать Удалить Отгрузить

Рисунок 16 – Процесс добавления груза на склад

Для того чтобы найти какой – то конкретный груз необходимо воспользоваться полем для поиска, введя соответствующее название груза (рисунок 17):

Система складского учета и логистики

Гофротара

Название

Артикул SKU

Количество

Цена

Масса в гр

Длина в см

Ширина в см

Высота в см

Добавить

Склад

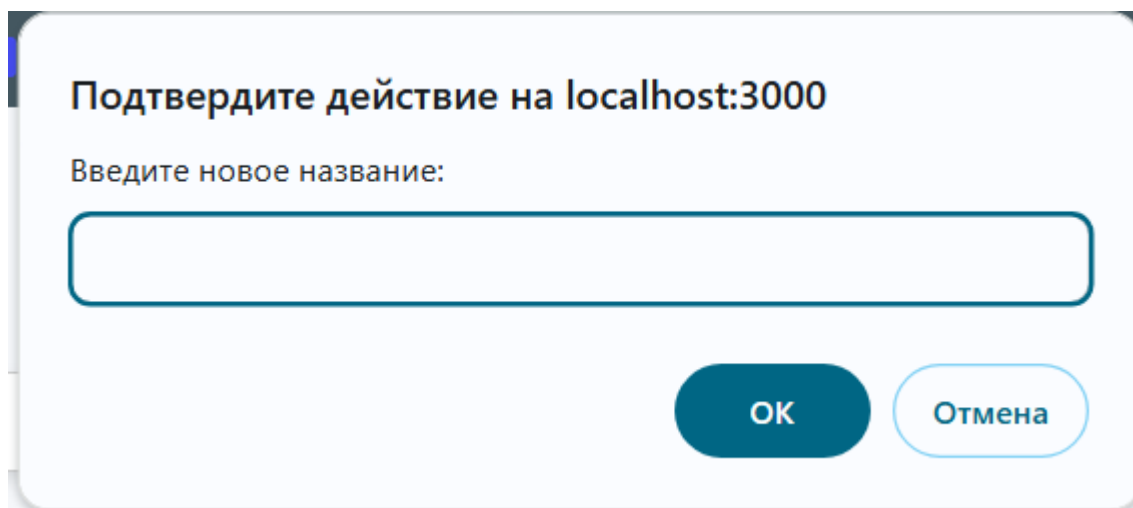
Название	Артикул SKU	Количество	Цена	Масса	Длина	Ширина	Высота	Дата поступления	Действия
Гофротара	Y523	15	130.00	241.00	15.00	15.00	10.00	06.01.2025, 23:06:27	Редактировать Удалить Отгрузить

История отгрузок

Груз	Количество	Артикул SKU	Дата отгрузки
------	------------	-------------	---------------

Рисунок 17 – Поиск груза

Записи о грузе на складе можно редактировать, если сотрудник допустил ошибку при заполнении полей. Для этого необходимо нажать на кнопку редактировать и на экране появится форма для редактирования для каждого из полей по очереди (рисунок 18):



Подтвердите действие на localhost:3000

Введите новое название:

OK Отмена

Рисунок 18 – Форма для редактирования

Также можно удалить запись о грузе на складе, если, например, она была добавлена ошибочно. Для этого следует нажать на кнопку удалить. После того как сотрудник нажмет на кнопку запись об определенном грузе пропадет.

Для наглядности создам 2 записи о грузах и удалю 1 из них (рисунки 19 – 20):

Склад								
Название	Артикул SKU	Количество	Цена	Масса	Длина	Ширина	Высота	Дата поступления
Гофротара	Y523	15	130.00	241.00	15.00	15.00	10.00	06.01.2025, 23:06:27
Гофрокороб	H123	20	150.00	374.00	17.00	17.00	14.00	06.01.2025, 23:19:37

Рисунок 19 – Создание записей о грузе

Склад								
Название	Артикул SKU	Количество	Цена	Масса	Длина	Ширина	Высота	Дата поступления
Гофротара	Y523	15	130.00	241.00	15.00	15.00	10.00	06.01.2025, 23:06:27
История отгрузок								

Рисунок 20 – Удаление записи

Кладовщик груз может выдать. Для этого необходимо нажать на кнопку отгрузить и в появившейся форме указать необходимое количество отгружаемого груза. Выдать груз можно частично или полностью. После того как кладовщик нажмет на кнопку отгрузить, выданный груз отобразится в истории отгрузок (рисунки 21 - 22):

Подтвердите действие на localhost:3000

Введите количество для отгрузки:

OK Отмена

Рисунок 21 – Форма для отгрузки груза

Склад									
Название	Артикул SKU	Количество	Цена	Масса	Длина	Ширина	Высота	Дата поступления	Действия
Гофротара	Y523	0	130.00	241.00	15.00	15.00	10.00	06.01.2025, 23:06:27	Редактировать Удалить Отгрузить
Гофрокороб	Z471	10	150.00	374.00	17.00	17.00	14.00	06.01.2025, 23:25:00	Редактировать Удалить Отгрузить

История отгрузок			
Груз	Количество	Артикул SKU	Дата отгрузки
Гофротара	15	Y523	06.01.2025, 23:29:47
Гофрокороб	10	Z471	06.01.2025, 23:30:24

Рисунок 22 – Отгруженный груз

Также, если кладовщик при указании количества отгружаемого груза допустит ошибку и укажет большее количество, чем то, сколько есть на складе, то получит сообщение о том, что груза на складе недостаточно (рисунок 23):

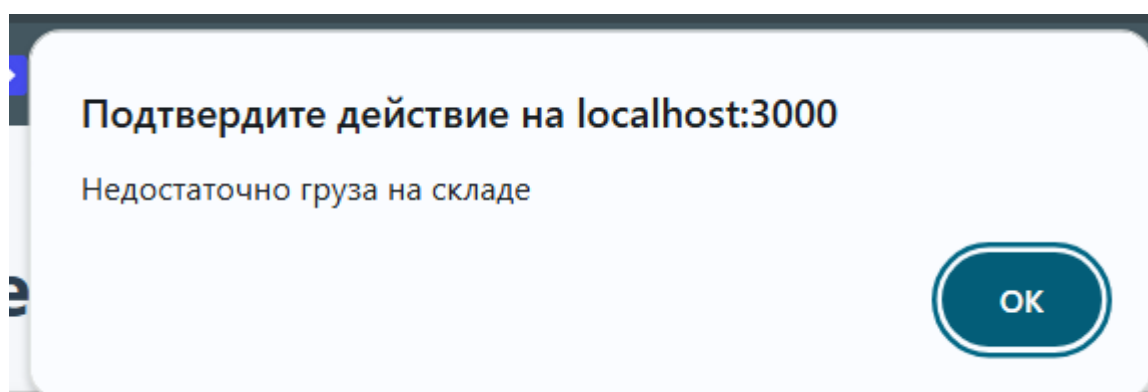


Рисунок 23 – Сообщение о недостатке груза

Также продемонстрирую печатную форму веб – приложения в виде отчета по остаткам и отчету по расходам. Для этого обращусь к базе данных и получу необходимую мне информацию с помощью SQL запросов, после чего сохраню информацию в виде Excel таблиц.

В качестве пример получу отчет по остаткам на 14.01.2025, воспользовавшись следующим SQL запросом (рисунки 24 - 25):

```

1  ▼ SELECT
2      name AS позиция,
3      quantity AS количество,
4      received_at::DATE AS дата
5  FROM
6      products
7  WHERE
8      received_at::DATE <= '2025-01-15'
9  ORDER BY
10     name;|

```

Рисунок 24 – SQL запрос для получения отчета по остаткам

	позиция character varying (100) 	количество integer 	дата date 
1	Гофрокартон 2-х слойный	8	2025-01-14
2	Гофрокартон 3-х слойный	40	2025-01-14
3	Гофрокартон 5-ти слойный	20	2025-01-14
4	Микрогофрокартон	33	2025-01-14

Рисунок 25 – Отчет по остаткам

Теперь выведу отчет по расходам гофрокартона 2-х слойного, также применив соответствующий SQL запрос (рисунки 26 - 27):

```

1  ✓ SELECT
2      shipped_at::DATE AS дата,
3      quantity AS количество,
4      purpose AS продукт
5  FROM
6      shipments
7  WHERE
8      sku = 'DF124'
9  ORDER BY
10     shipped_at;

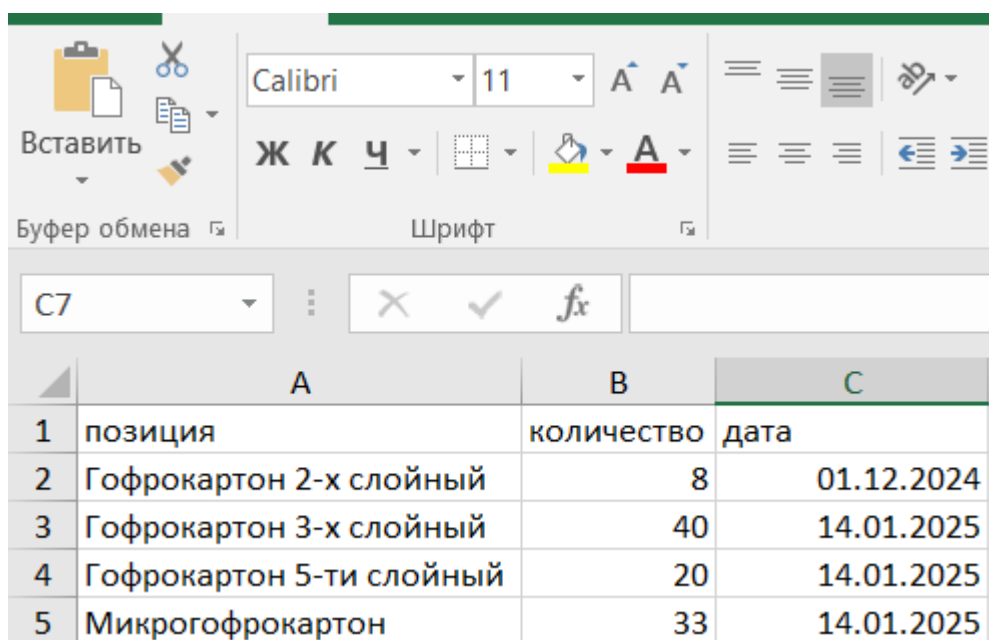
```

Рисунок 26 – SQL запрос для получения отчета по расходам

	дата date	количество integer	продукт character varying (255)
1	2024-12-05	5	гофротара
2	2024-12-10	7	гофрокороб
3	2024-12-15	10	гофроящик
4	2025-01-05	20	крупногабаритная упаковка

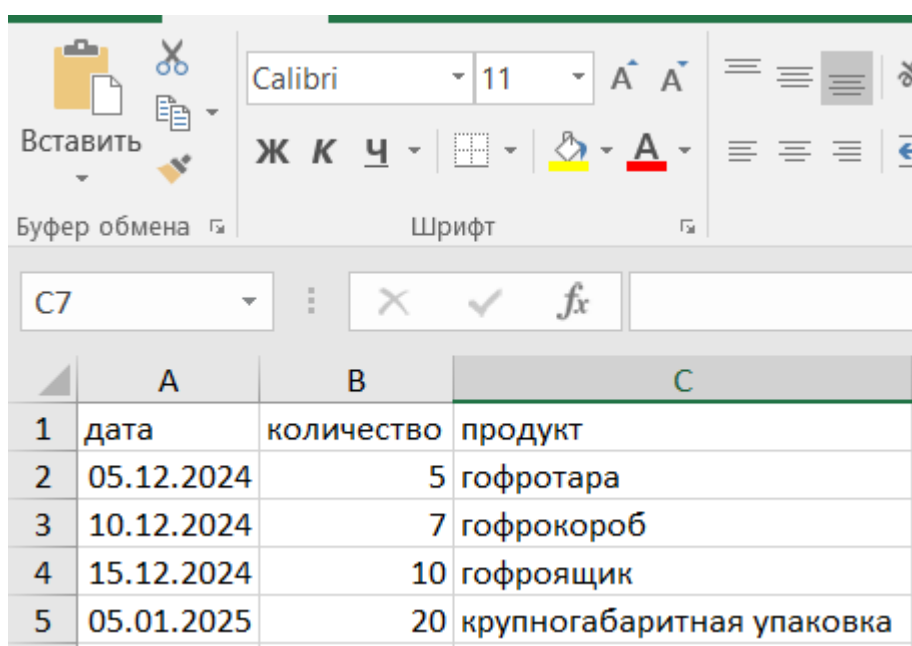
Рисунок 27 – Отчет по расходам

Получив отчеты, сохраню и представлю их в Excel таблицах. (рисунки 28 – 29):



	A	B	C
1	позиция	количество	дата
2	Гофрокартон 2-х слойный	8	01.12.2024
3	Гофрокартон 3-х слойный	40	14.01.2025
4	Гофрокартон 5-ти слойный	20	14.01.2025
5	Микрогофрокартон	33	14.01.2025

Рисунок 28 – Представление отчета по остаткам в Excel



	A	B	C
1	дата	количество	продукт
2	05.12.2024	5	гофротара
3	10.12.2024	7	гофрокороб
4	15.12.2024	10	гофроящик
5	05.01.2025	20	крупногабаритная упаковка

Рисунок 29 – Представление отчета по расходам в Excel

Данные из PostgreSQL успешно представлены в формате Excel, что обеспечивает их наглядное отображение и удобство для дальнейшего анализа.

3.3 Тестирование веб – приложения

Тестирование – важный этап разработки, направленный на проверку кода и выявление ошибок в работе системы [14]. Глубокое и качественное тестирование помогает устранить возможные проблемы и гарантировать надежность работы приложения. В качестве объекта тестирования используется разработанное веб-приложение внутренней логистики.

Приведу сценарии тестирования веб приложения.

В производстве нужно сделать 200 коробочек, для этого нужно 50 листов гофрокартона. Инженер, прежде чем выписывать накладную, убеждается, что на складе такой картон есть в нужном количестве (рисунок 30):

Склад							
Название	Артикул SKU	Количество	Цена	Масса	Длина	Ширина	Высота
Лист гофрокартона	G471	50	125.00	800.00	200.00	122.00	4.00

Рисунок 30 – Наличие достаточного количества гофрокартона

Водитель с накладной, чтобы получить 50 листов картона приезжает на склад, и кладовщик картон выдает и заносит данные в компьютер (рисунок 31):

История отгрузок			
Товар	Количество	Артикул SKU	Дата отгрузки
Лист гофрокартона	50	G471	06.01.2025, 22:09:51

Рисунок 31 – Занесение данных

Производство сделало 200 коробок. Инженер-производственник выписывает накладную на сдачу 200 коробок и отправляет водителя на склад. Кладовщик принимает 200 коробок (рисунок 32):

Коробка	G472	200	69.00	450.00	20.00	20.00	30.00	06.01.2025, 22:14:32
..								

Рисунок 32 – Принятие коробок на склад

Выводы по главе 3

В третьей главе подробно рассмотрена архитектура веб-приложения, включая ключевые компоненты, их взаимосвязь и принципы взаимодействия. Было уделено внимание как общей логике построения приложения, так и особенностям его реализации, что позволило раскрыть основные подходы, используемые при разработке. В рамках описания серверной части были рассмотрены механизмы обработки запросов, взаимодействия с базой данных, а также реализация маршрутизации и основных API - эндпоинтов.

Клиентская часть была проанализирована с точки зрения пользовательского интерфейса и обеспечения удобства взаимодействия для конечного пользователя. Описаны основные элементы интерфейса, принципы их организации и способы, которыми пользователи могут взаимодействовать с системой для выполнения задач.

Кроме того, глава включает демонстрацию ключевых функциональных возможностей приложения. Рассмотрены примеры работы основных функций, проанализированы сценарии их использования и представлена их важность в контексте общего функционала системы.

Завершает главу процесс тестирования веб-приложения.

Заключение

В ходе выполнения данной выпускной квалификационной работы было разработано веб – приложение для управления внутренней логистикой предприятия ООО «Волжский картон», что позволило автоматизировать ключевые бизнес-процессы предприятия и повысить их эффективность.

На первом этапе работы было проведено глубокое исследование предприятия и его организационной структуры, что дало понимание особенностей деятельности компании и определило цели автоматизации. Были разработаны модели «как есть» и «как должно быть» с использованием нотации IDEF0, что позволило визуализировать текущие процессы и выявить возможные пути их оптимизации. На основании анализа процессов были определены задачи и цели автоматизации, что послужило основой для дальнейшей работы над проектом.

В рамках второго этапа был выбран объектно-ориентированный подход к проектированию системы, который позволил разделить функциональность на модули и обеспечить гибкость и масштабируемость приложения. Проектирование логической и физической моделей данных позволило структурировать информацию и оптимизировать хранение данных, что способствует эффективной обработке информации и поддерживает высокую производительность системы. Было также предусмотрено удобное взаимодействие пользователей с системой, что повышает её доступность и интуитивную понятность.

На завершающем этапе работы было проведено архитектурное описание веб – приложения, демонстрация его возможностей, а также было проведено его тестирование.

В заключение можно сказать, что выполненная работа отвечает всем целям и задачам, поставленным в начале исследования.

Список используемой литературы

1. Бродский Б. Е. Моделирование бизнес процессов на предприятии. М.: Юнити, 2020. 270 с.
2. Брызгалин А.В. Проектирование информационных систем: архитектура и методы разработки. СПб.: БВХ-Петербург, 2021. 280 с.
3. Бугаев Ю. В., Коробова Л.А., Черняева С.Н. Исследование и моделирование информационных процессов и систем: учебное пособие Воронеж: Воронежский государственный университет инженерных технологий, 2022. 108 с.
4. Васильев Алексей Николаевич JavaScript в примерах и задачах. М.: Эксмо, 2022. 720 с.
5. Вейцман В. М. Проектирование информационных систем: учебное пособие для вузов. СПб.: Лань, 2022. 316 с.
6. Гаврилова Т. А., Кудрявцев Д. А., Блинов А. О. Моделирование бизнес – процессов. СПб.: Питер, 2022. 310 с.
7. Как работают веб – приложения / Хабр [Электронный ресурс] URL: <https://habr.com/ru/articles/450282/> (дата обращения: 01.11.2024).
8. Комаров В. И. Путеводитель по базам данных. М.: ДМК Пресс, 2024. 520 с.
9. Кузнецов А.А., Попов С. П. Организационное проектирование. СПб.: Питер, 2022. 320 с.
10. Мартин Роберт. Чистый код: создание, анализ и рефакторинг. СПб.: Питер, 2022. 464 с.
11. Мэтьюз Д. UML и паттерны проектирования. СПб.: Питер, 2021. 360 с.
12. Организационные структуры предприятия / Мое дело [Электронный ресурс] URL: <https://www.moedelo.org/club/upravlencheskiy-uchet/organizacionnyye-struktury-predpriyatiya> (дата обращения: 09.10.2024).
13. Рогов Е. В. PostgreSQL изнутри. М.: ДМК Пресс, 2024. 664 с.

14. Старолетов С.М. Основы тестирования программного обеспечения. СПб.: Лань, 2023. 192 с.
15. Требования к системе: классификация FURPS+ / WordPress [Электронный ресурс] URL: <https://sysana.wordpress.com/2010/09/16/furps/> (дата обращения: 14.01.2025).
16. Что такое UI – и UX – дизайн, чем они отличаются. Принципы и перспективы UI/ UX дизайна [Электронный ресурс] URL: <https://practicum.yandex.ru/blog/chto-takoe-ux-ui-dizayn/> (дата обращения: 02.12.2024).
17. Янкевич А.А. Системный анализ и проектирование информационных систем. Минск.: ТетраСистемс, 2022. 290 с.
18. Bass, L., Clements, P., & Kazman, R. Software Architecture in Practice (4th Edition). Boston: Addison-Wesley Professional, 2021. 640 p.
19. Connecting to Postgres with Nodejs / Tembo Docs [Электронный ресурс] URL: https://tembo.io/docs/getting-started/postgres_guides/connecting-to-postgres-with-nodejs (дата обращения: 18.11.2024).
20. Erl, T. Service-Oriented Architecture: Analysis and Design for Services and Microservices. Boston: Prentice Hall, 2020. 800 p.
21. Larman, C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition). Boston: Prentice Hall, 2020. 736 p.
22. Sommerville, I. Software Engineering. Harlow: Pearson Education, 2021. 816 p.