

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 /департамент /центр Прикладная математика и информатика

(наименование кафедры/департамента/центра полностью)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Бизнес-информатика

(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка автоматизированной системы управления торговым предприятием»

Обучающийся О.Ю. Балашова

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.п.н., доцент Т.А. Агошкова

(ученая степень, ученое звание, Инициалы Фамилия)

Тольятти 2023

## Аннотация

Название представленной выпускной квалификационной работы: Разработка автоматизированной системы управления торговым предприятием.

Цель выпускной квалификационной работы: создать автоматизированную систему управления торгового предприятия.

Объект выпускной квалификационной работы: система управления торговым предприятием.

Предмет исследования: автоматизация информационных процессов в управлении торгового предприятия.

Структурно работа представлена введением, четырьмя разделами, заключением и списком использованных источников.

В первой главе представлен анализ объекта, приведено описание предметной области, сделано построение концептуальной модели предметной области.

Во второй главе приведено описание аналогов предметной области, сделан обзор и обоснование выбора инструментарий разработки и разработано техническое задание.

В третьей главе представлена разработка архитектуры программного продукта, сделано проектирование структур хранения данных и описание реализации вариантов использования.

В четвертой главе приведена разработка алгоритмов реализации вариантов использования, классов информационной системы и графического интерфейса, а также приведено тестирование программной системы, выполнено экономическое обоснование проекта.

Пояснительная записка содержит 61 листа печатного текста, 41 иллюстрацию, 17 таблиц, 11 формул. Пояснительная записка состоит из введения, аннотации, 4 главы, заключения и списка использованных источников, включающий 20 наименований.

## Оглавление

Введение.....	4
Глава 1 Анализ объекта.....	7
1.1 Описание предметной области.....	7
1.2 Построение концептуальной модели предметной области.....	9
Глава 2 Постановка задачи определение требований к программной системе .....	10
2.1 Описание аналогов предметной области .....	10
2.2 Обзор и обоснование выбора инструментарий разработки .....	15
Глава 3 Проектирование программной системы.....	18
3.1 Разработка архитектуры программного продукта .....	18
3.2 Проектирование структур хранения данных .....	32
3.3 Описание реализации вариантов использования .....	36
Глава 4 Реализация программной системы.....	39
4.1 Разработка алгоритмов реализации вариантов использования	39
4.2 Разработка классов информационной системы.....	42
4.3 Разработка графического интерфейса .....	45
4.4 Тестирование программной системы .....	49
4.5 Экономическое обоснование проекта.....	53
Заключение .....	59
Список используемой литературы .....	61

## Введение

Успешный бизнес во многом похож на езду на велосипеде: он требует баланса, прогресса и правильных инструментов. Когда дело доходит до управления жизненным циклом доставки программного обеспечения команды и управления жизненным циклом приложений, автоматизация обеспечивает скорость, гибкость и надежность, необходимые для сохранения баланса и удовлетворения постоянно меняющихся требований клиентов.

Автоматизация процессов — немалая задача, но при наличии правильной стратегии и инструментов она может стать эффективным способом оптимизации циклов разработки и обеспечения контроля качества.

Программное обеспечение для автоматизации бизнеса — это набор инструментов, которые помогают автоматизировать повторяющиеся бизнес-операции, оптимизировать рабочие процессы и повысить эффективность бизнес-процессов. К тому же, возможности электронной отчетности и электронного документооборота снимают большую нагрузку с руководителей и сотрудников финансовых отделов предприятия, позволяя вести отчетность «одним нажатием кнопки» и не требующее длительного ручного труда по сверке правильности подсчетов.

Программное обеспечение для автоматизации бизнеса может быть либо универсальной платформой, которая помогает вам автоматизировать различные бизнес-операции из одного места, либо автономным решением для автоматизации, предназначенным для автоматизации конкретной бизнес-операции. Современные возможности позволяют работать в системе через облачные решения, если предприятие не может позволить себе установку дорогостоящего оборудования.

Несмотря на то, что автоматизация помогает избавиться от повторяющихся задач, 40 % автоматизации сегодня по-прежнему не имеют структуры, потому что предприятия не выделяют достаточно денег, чтобы тратить их на программное обеспечение для автоматизации бизнеса. Это

большая финансовая затрата, которую не каждая компания может себе позволить. Большинство руководителей предприятий требуют обоснование, которое должно показать успех «здесь и сейчас», и даже не рассматривают то, что может сократить расходы или даже принести прибыль спустя некоторый период времени.

В некоторых случаях компании готовы пойти на затраты финансов на приобретение простой и недорогой системы автоматизации. Чаще всего подобные варианты решений будут иметь урезанный функционал, расширение которого возможно будет за дополнительную оплату. Или же система будет иметь возможность работать только с одним направлением бизнеса и при данном варианте нужно будет приобретать для каждого направления отдельную систему и пытаться объединить их подручными средствами. Это означает, что такая система не будет соответствовать потребностям и ожиданиям сотрудников и клиентов, что затрудняет достижение желаемых бизнес-результатов.

И если нужно масштабировать бизнес-операции, повысить производительность сотрудников или повысить качество обслуживания клиентов, нужно выйти за рамки базовой автоматизации и подумать об автоматизации рабочих процессов. Ведь автоматизация позволит сократить участие человека в бизнес-процессах, что в свою очередь снижает процент возникающих ошибок в связи с «человеческим фактором».

Объект выпускной квалификационной работы: система управления торговым предприятием.

Предмет исследования: автоматизация информационных процессов в управлении торгового предприятия.

Цель выпускной квалификационной работы: создать автоматизированную систему управления торгового предприятия (движение товаров на складе).

Для этого необходимо решить следующие задачи:

- проанализировать организационную структуру,

- проанализировать функционирование,
- проанализировать использование информации,
- спроектировать базу данных,
- разработать оконный графический интерфейс для базы данных.

В качестве метода исследования бизнес-процессов был выбран метод моделирования бизнес-процессов. Моделирование бизнес-процессов позволяет исследовать не только цельную картину функционирования организации, ее взаимодействия с внешними предприятиями, клиентами, поставщиками, но и как организована деятельность внутри организации на каждом отдельно взятом рабочем месте. В данной выпускной квалификационной работе был выбран графический способ описания и моделирования бизнес-процессов. Графическое моделирование использует стандартизированные схемы и позволяет визуализировать данные на всех этапах – от сбора информации построения организационной структуры до согласования с заказчиком этапов бизнес-процессов и их внедрения. В качестве инструмента был использован язык графического описания для объектного моделирования в области разработки программного обеспечения – UML.

Результатом выпускной квалификационной работы станет программное средство для торгового предприятия.

Структурно выпускная контрольная работа представлена аннотацией, введением, пятью главами, заключением и списком использованных источников.

## **Глава 1    Анализ объекта**

### **1.1    Описание предметной области**

База данных разрабатывается для торгового предприятия - магазина. Она предназначена для информационного обслуживания сотрудников склада и магазина. БД должна содержать данные о товарах, продажах, учет на складе, контроль продаж сотрудниками и предоставлять возможность получать разнообразные отчеты.

Магазин осуществляет продажу одежды от поставщика к конечному потребителю. Какой набор услуг включены в сферу интереса магазина одежды:

- покупка одежды у поставщиков,
- продажа одежды покупателю,
- поиск поставщиков и покупателей,
- формирование прозрачной отчетности.

В организационной структуре предприятия три отдела и системный администратор, находящийся в подчинении непосредственно директора. Старший продавец ведет учет товара, занимается его продажей, наблюдает за работой продавцов и контролирует чистоту помещений, а также рекламой товара магазина. Главный бухгалтер занимается проведением взаиморасчетов между поставщиками и магазином, а контролирует правильность проведения покупок товара. Начальник склада занимается контролем движения товара на складе. Организационная схема магазина показана на рисунках 1 и 2.

Исходя из вышесказанного, выделим базовые сущности предметной области:

- поставщики,
- продажа,
- склад,
- сотрудники,

- товары,
- рассылки,
- заказы расходных материалов.



Рисунок 1 - Организационная схема



Рисунок 2 - Организационная схема по должностям

На основе полученной информации можно сделать вывод о том, что организационная структура данного предприятия является линейно-функциональной. Результаты вывода потребуются для дальнейшей разработки автоматизированной системы.



## 1.2 Построение концептуальной модели предметной области

База данных для предприятия создается для обслуживания следующих групп пользователей:

- сотрудник склада,
- продавец,
- директор,
- бухгалтер,
- системный администратор.

Проектируемая БД сможет обеспечивать функциональные возможности, такие как:

- ведение БД (запись, чтение, модификация, удаление в архив);
- обеспечение целостности и логической непротиворечивости данных;
- обеспечение защиты данных от несанкционированного или случайного доступа (определение прав доступа);
- реализация наиболее часто встречающихся запросов в готовом виде.

Построение концептуального моделирования - необходимый этап в разработке информационной системы. Он позволит определить будущую структуру базы данных.

## Глава 2 Постановка задачи определение требований к программной системе

### 2.1 Описание аналогов предметной области

Помимо избавления от повторяющихся задач за счет автоматизации, потребности клиентов и сотрудников меняются, что приводит к увеличению использования программного обеспечения для автоматизации бизнеса.

Некоторые компании стремятся использовать одну платформу автоматизации для повышения эффективности своих бизнес-процессов. Другие ищут программное обеспечение для автоматизации бизнеса, которое поможет им улучшить качество обслуживания клиентов, в то время как другие хотят улучшить качество обслуживания сотрудников.

Рассмотрим некоторые готовые программные решения для ведения бизнеса.

Приведем сравнение аналогов на рисунке 3.

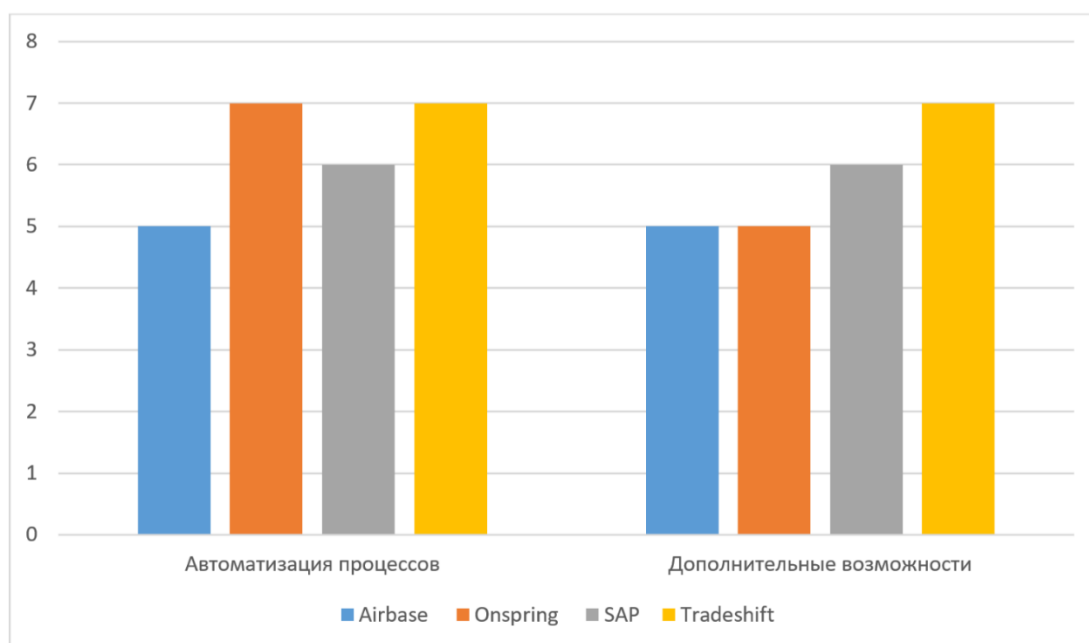


Рисунок 3 – Диаграмма сравнения аналогов

Airbase. Это решение для управления расходами, корпоративной карты и счетов оплаты, является универсальной платформой для управления расходами. ПО охватывает видимость компании и контролирует каждую расходную денежную единицу.

Платформа Airbase решает множество задач, таких как:

- запросы на расходы,
- одобрения,
- оплата счетов.

Он также автоматизирует кодирование и размещение в главной книге:

- основные характеристики,
- отчеты о расходах.

Создание отчетов о расходах, связанных с расходами на малое время. Он включает в себя управление цифровыми квитанциями, интеграцию с банком/кредитной картой и возмещение расходов.

Платежи могут обрабатываться партиями без ручного вмешательства. Автоматизируйте рабочие процессы и создавайте порталы самообслуживания, чтобы легко получать счета на обработку. Сверка и регулируемые счета-фактуры, платежи и банковские выписки.

Airbase предлагает рекомендации по использованию иностранных языков. Расходы по проекту на основе подписанных контрактов и выявление избыточных транзакций.

Дополнительные возможности:

- управление контрактами с отслеживанием цен и каталогом приложений;
- независимая интеграция с другими продуктами;
- защита от мошенничества с помощью SSL-шифрования и соответствия стандарту PCI;
- способы оплаты включают АСН, кредит/дебет и банковский перевод;

– признание дохода с отслеживанием использования и отложенным доходом.

Onspring. Облачная система управления бизнесом. Выявленное вознаграждение за решение проблемы с обнаружением автоматизации и отчетами в режиме реального времени. Данная система является гибкой, масштабируемой и подходит для таких презентаций, как управление рисками, управление и аудит.

Компания верит в «автоматизацию, управляемую людьми» и помогает бизнесу использовать технологии, которые улучшают их системы управления и упрощают рабочие процессы. Платформу можно настроить в соответствии с потребностями каждой команды, чтобы помочь вам работать умнее, а не усерднее. Приведем основные характеристики.

Возможность автоматизации бизнес-процессов позволяет управлять полным жизненным циклом поставщика с закупками и проводить стандартизированный процесс адаптации. Выпуск индивидуальных оценок рисков поставщиков и отслеживание планов по снижению рисков. Определение поставщиков-дубликатов или повторяющихся поставщиков по всему предприятию. Контроль доступ по группам, пользователям или ролям.

Отмеченное наградами решение полностью интегрирует стратегию управления рисками, объединяя политики, риски и соответствие требованиям в одном простом инструменте. Это охватывает такие задачи, как:

- внутренний аудит и гарантии,
- непрерывность бизнеса и восстановление,
- управление инцидентами и политиками.

Эффективное управление инцидентами, контрактами и непрерывностью с помощью надежных рабочих процессов, отчетов в режиме реального времени и элементов управления на основе ролей. Это делает деловые операции более продуктивными.

Onspring — это самый быстрый способ управлять нормативными актами и соответствовать постоянно меняющимся нормативным стандартам.

Дополнительные возможности:

- аналитика с отчетами в реальном времени,
- автоматизация динамических рабочих процессов, сообщений и интеграций,
- пользовательские информационные панели,
- полная консолидация данных с расширенными формулами,
- административная поддержка с обучением и помощью в реализации.

SAP. Являясь ведущим мировым брендом в сфере ИТ, предлагает облачное решение для совместного управления расходами, продажами и финансами. Хотя системы CRM и ERP великолепны, они мало что делают для улучшения взаимодействия между вами и вашими поставщиками.

Обычно это по-прежнему управляется с помощью фрагментированного набора приложений и ручных процессов. Это не только увеличивает затраты, но и блокирует ваши просмотры и представляет более высокий риск. SAP упрощает и повышает эффективность сотрудничества с поставщиками, решая все эти проблемы.

Разработка является преимуществом в управлении рисками поставщиков. Когда покупатель хорошо информирован, он может избежать принятия решений, создающих больший риск для бизнеса. Жизненный цикл и производительность SAP помогает централизованно управлять всем жизненным циклом ваших поставок, от регистрации до поэтапного отказа. Использование информации о поставщиках, основанную на искусственном интеллекте, позволит постоянно повышать эффективность расходов.

Благодаря сотрудничеству и обратной связи с поставщиками в режиме реального времени SAP экономит ваши трудозатраты. Аналитические данные предоставляются платформой управления опытом Qualtrics.

Дополнительные возможности:

- выбор поставщиков с использованием актуального профиля рисков с указанием категории, местоположения, бизнес-подразделения;

- переговоры по контракту, установление условий и KPI риска;
- единое представление поставщика на основе матрицы для управления производительностью и снижения рисков;
- адаптация с пониманием оценки рисков и статуса аудита;
- источник и управление прямыми и косвенными материалами на единой платформе.

Tradeshift. Является комплексной платформой для ведения бизнеса. Это сетевое решение №1 для автоматизации процессов поставщиков. Система позволяет компании работать без бумаги и бесконтактно на протяжении всего цикла «от заказа до оплаты».

Программа была разработана, чтобы помочь бизнесу расти за счет создания современной сети цепочки поставок B2B и оцифровки торговли. Tradeshift помогает бренду постоянно адаптироваться, независимо от скорости изменений.

Tradeshift предоставляет компании бесплатный доступ к сотрудничеству с поставщиками и аналитике. С первого дня можно привести продавцов к более продуктивному партнерству.

Возможность привлекать больше ценностей с помощью практических идей, которые позволяют принимать более эффективные решения. Это включает в себя инструменты для лучшей видимости транзакций, расширенной аналитики и сбора данных из вашего решения ERP.

Общение между покупателями и продавцами никогда не было проще. Все взаимодействия на платформе Tradeshift отслеживаются и документируются. Вам больше не придется рыться в почтовом ящике.

Избавление от бумажных квитанций и чрезмерных расходов персонала. Tradeshift Go — это веб-приложение, которое делает корпоративные карты лучше. Дополнительные возможности:

- Tradeshift Pay совершенствует процессы AP и автоматизирует платежные циклы;

- гарантированный уровень адаптации поставщиков с реальной ценностью для привлечения поставщиков;
- панель управления с улучшенным ИИ, которая переключается вверх/вниз в зависимости от желаемого элемента управления;
- Tradeshift Buy, чтобы заполнить вашу цепочку поставок предложениями от выбранных поставщиков.

После проведения сравнения по возможностям автоматизации и дополнительным возможностям было выявлено, что не все предложенные варианты могут обеспечить необходимый функционал. В случае, когда ПО было максимально близко к требованиям, присутствовали лишние возможности, которые удорожали систему. Было принято решение о разработке собственной информационной системы с необходимым функционалом.

## **2.2 Обзор и обоснование выбора инструментарий разработки**

Microsoft SQL Server - это система управления реляционными базами данных (RDBMS), разработанная Microsoft. По сути, это программное обеспечение, позволяющее эффективно хранить, изменять и управлять реляционными данными. Наряду с Oracle, MySQL, PostgreSQL и DB2 Microsoft SQL Server является одной из самых популярных систем управления реляционными базами данных на рынке [19].

Одной из наиболее важных функций SQL Server является предоставление данных другим программным приложениям (клиентам). Чтобы взаимодействовать с базами данных SQL Server и управлять их данными или запрашивать их, вы можете использовать язык Transact-SQL (T-SQL), расширение стандарта SQL.

Преимущество Microsoft SQL Server [17]:

- повышение безопасности данных: обеспечивает безопасность базы данных, особенно с помощью службы администратора базы данных MS-SQL Server;
- простота настройки: в отличие от других систем управления базами данных установка и настройка проще;
- оптимизированное хранилище данных: нам не требуется какое-либо другое хранилище данных для той же базы данных при использовании другого устройства. Это также позволяет легко устранять неполадки и поддерживать данные;
- поддержка восстановления данных: в случае сбоя питания или отключения сервера данные могут быть повреждены, поэтому Microsoft SQL Server устраняет риск потери данных благодаря функциям восстановления и восстановления данных.

#### Недостатки Microsoft SQL Server [20]:

- стоимость: если вам нужны дополнительные функции и приложение базы данных, вы также можете инвестировать в более высокую версию;
- ограниченная совместимость: вы можете сделать некоторые дополнительные инвестиции в программное обеспечение Microsoft, если хотите использовать небольшую инфраструктуру Microsoft, но они позволят вам использовать Microsoft SQL Server на вашей платформе;
- ограничение по оборудованию: вам может понадобиться более новый компьютер, если у вас более старая версия оборудования, потому что для работы более новых версий Microsoft SQL Server требуются передовые технологии [18].

Visual Studio — это решение для разработки приложений, которое помогает компаниям создавать приложения и игры для Интернета, мобильных устройств, Mac и Windows [7]. Разработчики могут оптимизировать процессы отладки, тестирования и развертывания на централизованной платформе. Visual Studio имеет встроенные инструменты тестирования и шаблоны



проектов, которые позволяют компаниям писать, запускать и отлаживать модульные тесты на нескольких языках и платформах тестирования.

Сотрудники могут использовать это решение для навигации по коду, выявления ошибок и внесения исправлений в код.

Администраторы могут использовать модуль Live Sharing для настройки сеансов совместного редактирования и отладки кода с правами доступа и настраиваемыми настройками редактора для обеспечения соблюдения predetermined рекомендаций по написанию кода.

Кроме того, продукт предоставляет несколько сочетаний клавиш для таких операций, как поиск, навигация, редактирование, рефакторинг и тестирование. Visual Studio интегрируется с различными сторонними решениями, такими как ReSharper, CodeMaid и GitHub Extension. Цены предоставляются по запросу, а поддержка осуществляется через форум сообщества, раздел часто задаваемых вопросов, документацию, чат и другие онлайн-ресурсы.

## Глава 3 Проектирование программной системы

### 3.1 Разработка архитектуры программного продукта

Организации используют моделирование бизнес-процессов (моделирование бизнес-процессов) для визуального документирования, понимания и улучшения своих процессов. Являясь частью управления бизнес-процессами, моделирование бизнес-процессов использовалось в качестве организационного инструмента для определения того, что есть (или «как есть») в качестве базовой линии, и для определения будущего (или «будущего») с любыми улучшениями усвоены [13].

Диаграмма модели бизнес-процесса — это инструмент - средство для достижения цели, а не результат производительности сам по себе. Именно она помогает понять дальнейшие шаги в работе, построение порядка действий процесса, возможность избежать лишние и промежуточные этапы, наглядно показать ключевые действия. Это один из лучших инструментов для визуализации бизнеса.

Конечным результатом является улучшение работы бизнес-процесса. Возможность оптимально распределить нагрузку между сотрудниками дает возможность избежать лишние затраты.

Основное внимание в улучшениях уделяется действиям с добавленной стоимостью, которые улучшают качество обслуживания клиентов и сокращают потери времени и усилий.

Существует два основных типа моделей бизнес-процессов:

- модель «AS-IS» или базовая модель (текущая ситуация),
- модель «TO-BE» (предполагаемая новая ситуация).

Используются для анализа, тестирования, внедрения и улучшения процесса.

Приведем перечень наиболее популярных техник, которые могут быть автономными или даже дополнять более формальные подходы:

- техника блок-схемы,
- диаграммы потоков данных - метод Юрдона,
- диаграммы ролей и действий (RAD),
- диаграммы взаимодействия ролей (RID),
- диаграмма Ганта,
- интегрированное определение для моделирования функций (IDEF),
- цветные сети Петри (CPN),
- объектно-ориентированные методы (ОО),
- техника рабочего процесса,
- моделирование,
- нотация моделирования бизнес-процессов (BPMN),
- диаграмма активности UML,
- модели трансформационных процессов,
- рассказывание историй,
- иерархические модели процессов,
- визуализация.

Для выполнения задания в выпускной квалификационной работы применим подход интегрированного определения для моделирования функций (IDEF) в программе AllFusion Process Modeler [8].

В данной работе на основе нотации IDEF0 была разработана контекстная диаграмма, которая показывает входные и выходные ресурсы, правила управления и механизм управления склада до внедрения информационной системы или модель «AS-IS» (рисунок 4).

Контекстная диаграмма верхнего уровня модели AS-IS разделена на пять процессов (Приемка товара, Хранение товара, Комплектование товара по требованию, Отгрузка товара и Проведение инвентаризации), как показано на рисунке 5, образующих первый уровень модели AS-IS.

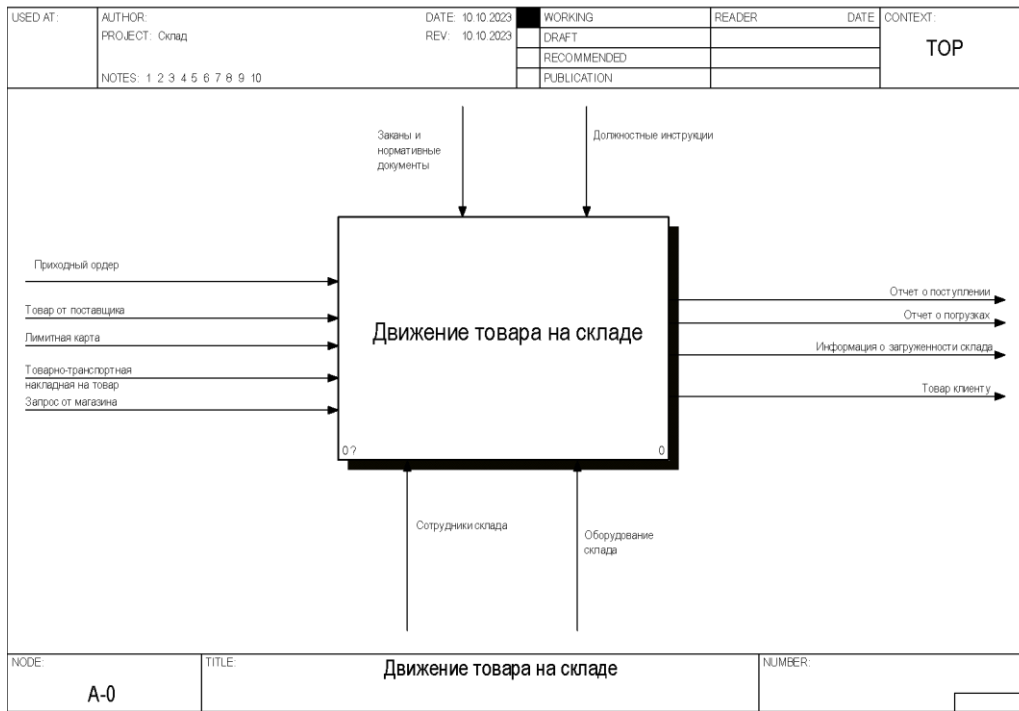


Рисунок 4 – Контекстная диаграмма функциональной модели системы в нотации IDEF0 модели «AS-IS»

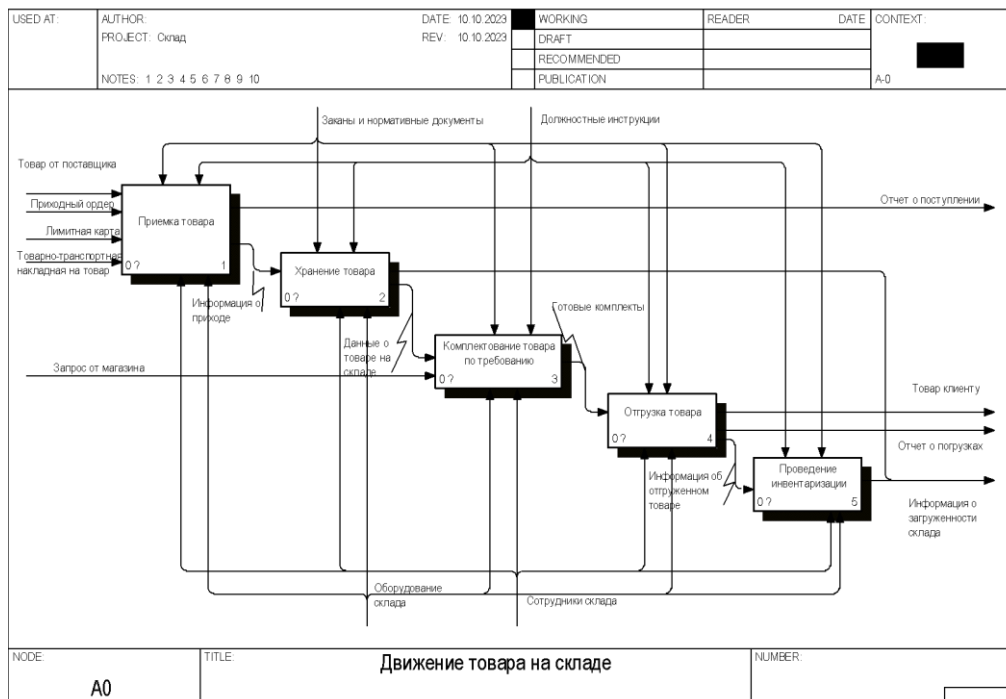


Рисунок 5 – Первый уровень функциональной модели системы в нотации IDEF0 модели «AS-IS»

Основываясь на данном построении, будет возможно более детально рассмотреть каждый процесс поэтапно и выстроить модель «как должно быть». Бизнес-процессы, представленные в рамках данной диаграммы представлены далее.

Бизнес-процесс «Приемка товара» (рисунок 6) описывает процесс поступления товара на склад. Этот бизнес-процесс можно разделить на три части: Приемка и проверка документов, Разгрузка товара и Проверка количества и качества товара. На данном уровне построена модель начального движения процесса, когда товар приходит в организацию и проходит первичные этапы от доставки до сверки соответствия документации. Ключевым моментом данного этапа является правильность заполнения документации о приемке. Малейшие ошибки могут привести к проблемам актуальности информации и материальным убыткам. Каждая часть данного бизнес-процесса сводит возможность возникновения ошибок к минимуму. Данный бизнес-процесс представлен ниже:

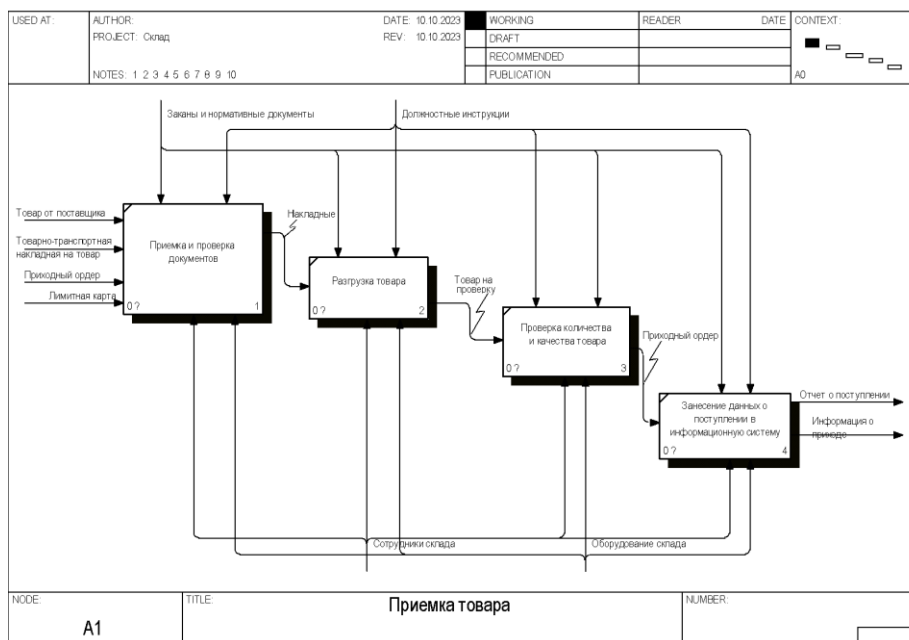


Рисунок 6 – Второй уровень функциональной модели системы в нотации IDEF0 модели «AS-IS»

Бизнес-процесс (рисунок 7) «Хранение товара» — это процесс хранения продукции на складе. Этот бизнес-процесс можно разделить на четыре части (Определить место хранения товара, Перемещение товара на место хранения, Размещение товара на месте хранения и Контроль наличия запасов). Данный уровень является следующим логическим этапом в работе с товаром. После поступления товара его необходимо разместить. Особенно важен этот этап на предприятиях, которые имеют большие складские помещения, многоуровневые хранилища или даже удаленные склады. Точная информация о местонахождении товара позволяет оперативно рассчитать дальнейшие действия по реализации. Функциональная модель процесса Хранение товара представлена далее:

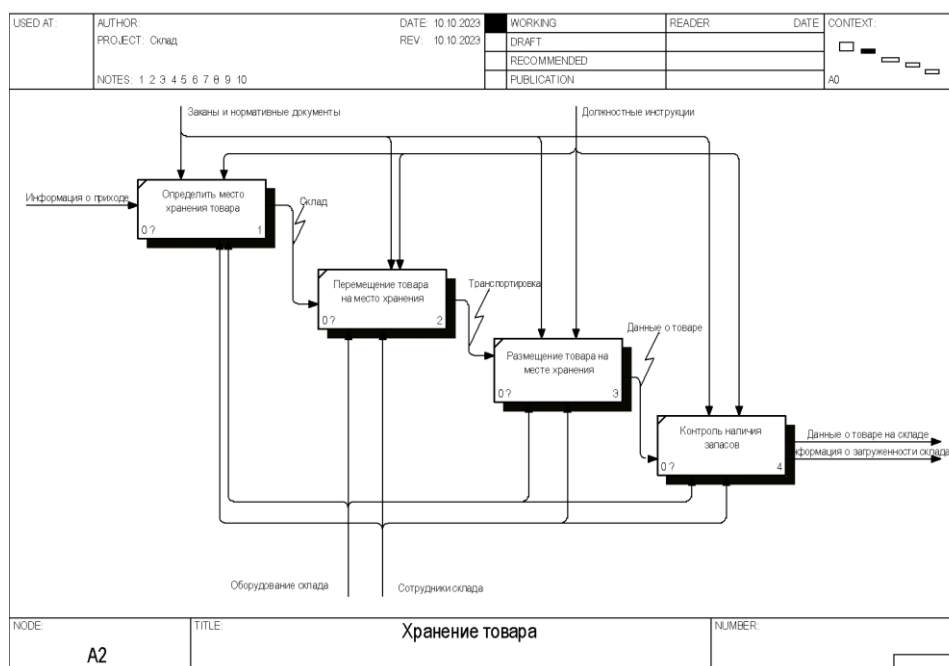


Рисунок 7 – Второй уровень функциональной модели системы в нотации IDEF0 модели «AS-IS»

Бизнес-процесс (рисунок 8) «Комплектования товара по требованию» включает в себя процесс сборки продукции по заказам. Он разделен на 5 шагов (Получение документов по заказам, Выборка товара согласно заказу,

Комплектование заказа, Подготовка сопроводительных документов и Запаковка и опломбировка товара). Самый обширный бизнес-процесс из представленных функциональных моделей торгового предприятия. На данном этапе задействованы сотрудники, которые формируют документацию и которые осуществляют комплектацию заказа. Внимательность сотрудников является ключевым фактором успешности продажи товара непосредственному покупателю. Процесс комплектования товара представлен далее:

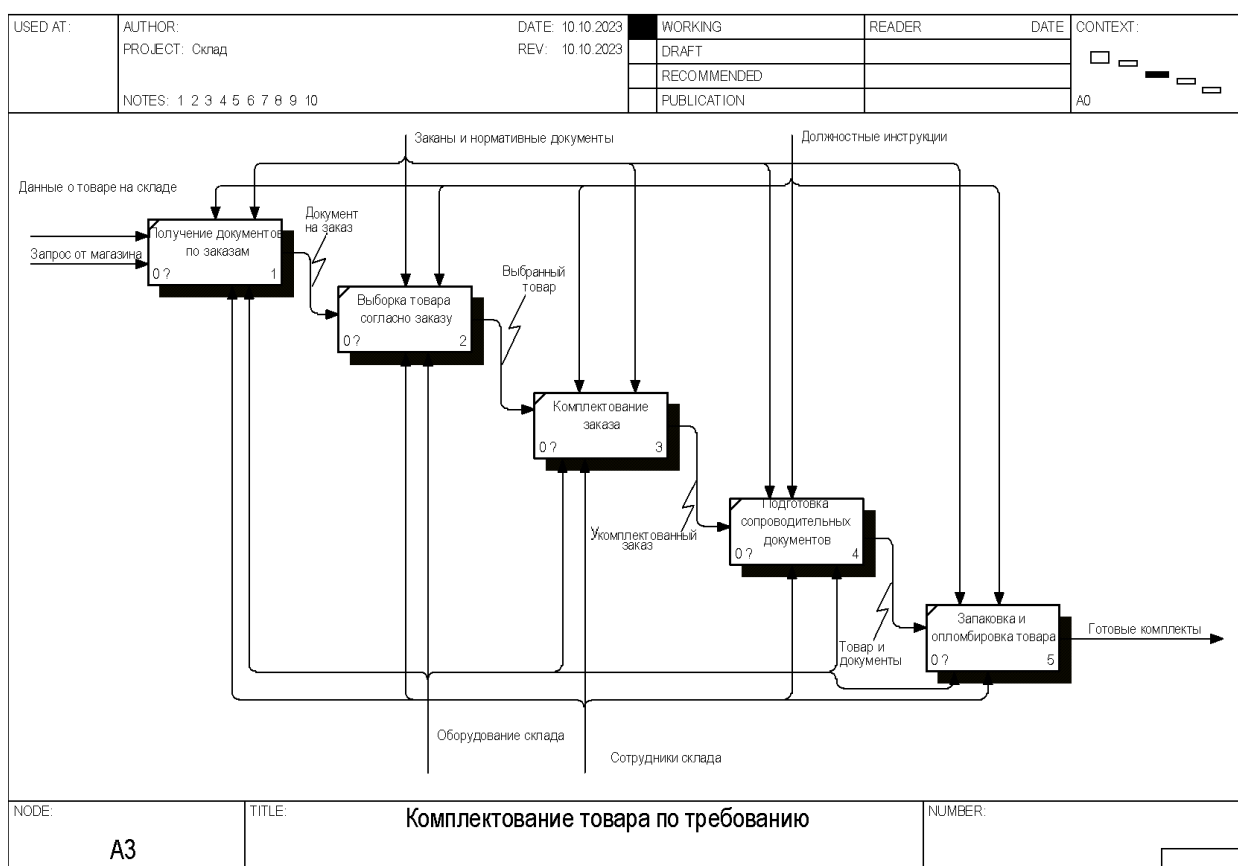


Рисунок 8 – Второй уровень функциональной модели системы в нотации IDEF0 модели «AS-IS»

Бизнес-процесс «Отгрузка товаров» (рисунок 9) предполагает доставку товаров покупателям. Этот бизнес-процесс можно разделить на три части (Оптимальное комплектование готовых заказов, Оформление транспортных

накладных и Погрузка товара в транспортное средство). Благодаря четкому выполнению действий данного бизнес-процесса удастся избежать проблем с пересортировкой или нарушением заказа. Что в свою очередь помогает сократить количество претензий и возвратов товара от покупателя. Более детально с построенным бизнес-процессом функциональной модели можно ознакомиться далее:

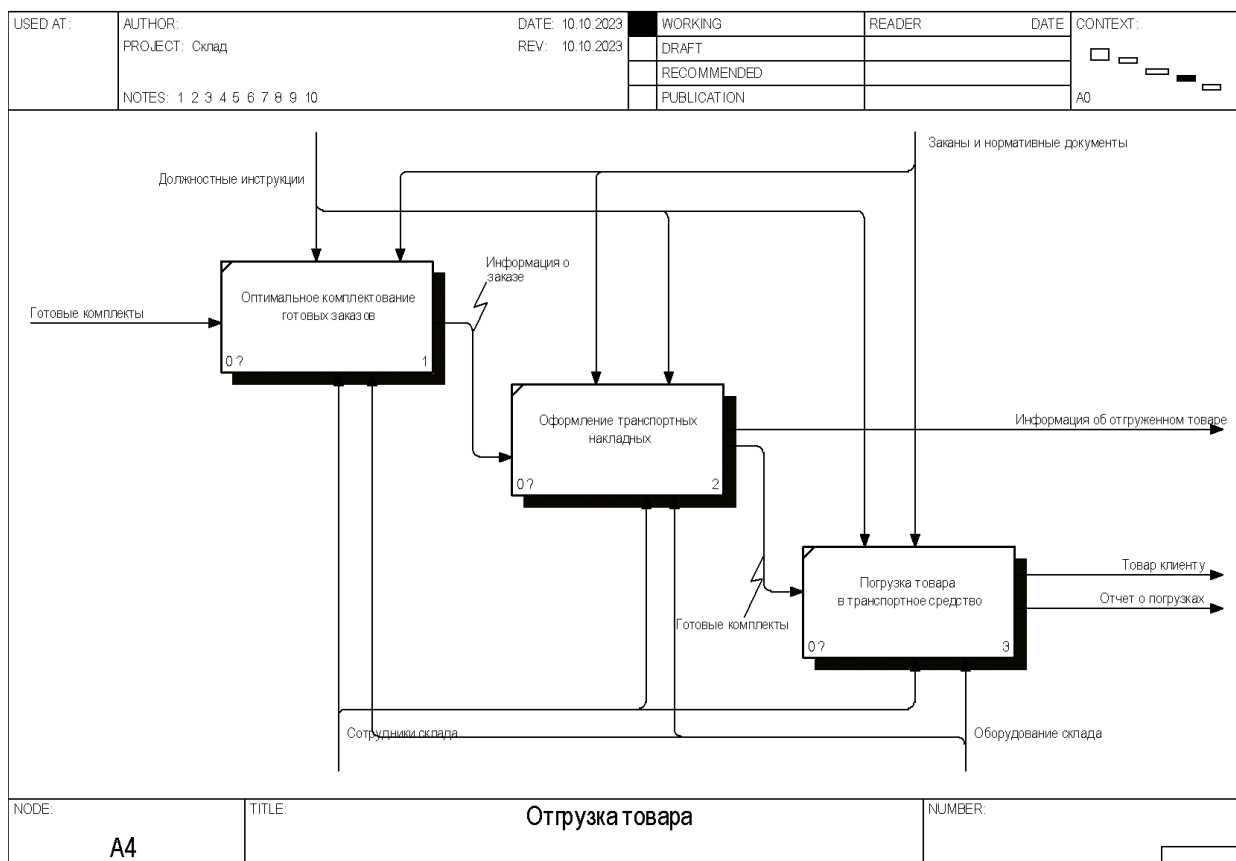


Рисунок 9 – Второй уровень функциональной модели системы в нотации IDEF0 модели «AS-IS»

Бизнес-процесс (рисунок 10) «Проведение инвентаризации» включает в себя процедуры проведения инвентаризации на складе. Этот бизнес-процесс можно разделить на три части (Подготовка документов о наличии товара на складе, Сверка наличия товара с документами и Подготовка отчета о результатах инвентаризации). Данный этап позволяет актуализировать



информацию о соответствии имеющегося товара в наличии с документацией прихода и расхода товара, составив тем самым полный отчет по передвижениям товара. Актуальность своевременного проведения инвентаризации подтверждается постоянным контролем над бизнес-процессами и позволяет избегать недостачи и убытки предприятия.

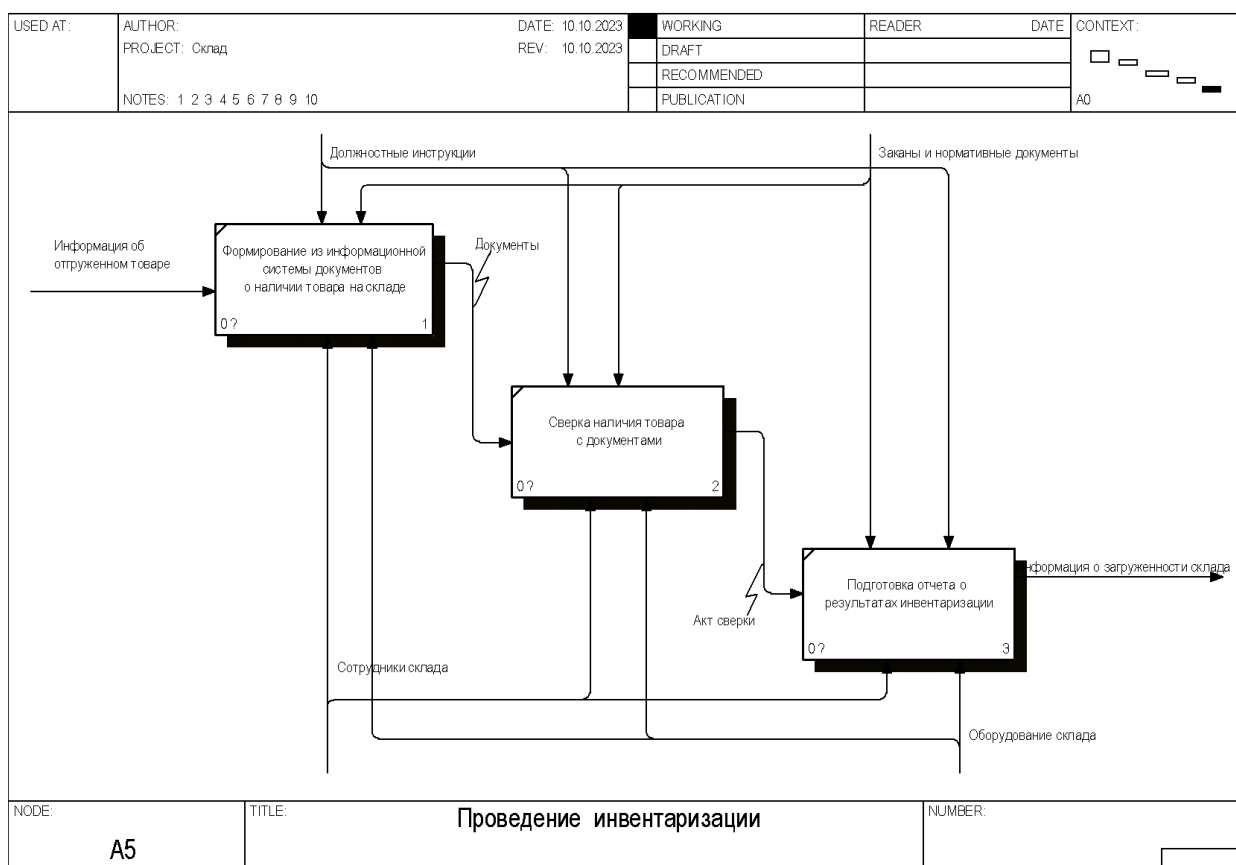


Рисунок 10 – Второй уровень функциональной модели системы в нотации IDEF0 модели «AS-IS»

Управление складом может быть важной и сложной задачей, особенно для предприятий, которые полагаются на эффективные и действенные складские операции для поддержки своей цепочки поставок. Эти проблемы могут существенно повлиять на прибыль бизнеса за счет увеличения затрат, снижения эффективности и негативного влияния на удовлетворенность клиентов [1]. Для предприятий важно уделять приоритетное внимание

эффективному управлению складами, чтобы обеспечить бесперебойную и структурированную работу. Решая эти проблемы и внедряя эффективные методы управления складами, предприятия могут улучшить свою деятельность и лучше удовлетворять потребности своих клиентов. Приведем некоторые распространенные проблемы, которые могут возникнуть при управлении складом.

Рабочие являются важным аспектом склада, и правильное управление ими может повысить общую производительность различных складских процедур. Управление большой рабочей силой может оказаться затруднительным, поскольку работники варьируются от помощников до менеджеров, а также при наличии дорогого оборудования. Большинство складов неэффективно организованы, что приводит к нерациональному использованию площадей и необходимости использования дополнительной рабочей силы для компенсации неэффективности. Другой причиной может быть отсутствие адекватных технологий, что может привести к неэффективности процессов и необходимости привлечения большего количества рабочей силы для ручного управления задачами, которые можно автоматизировать.

Руководство склада может столкнуться с проблемами контроля ущерба по ряду причин. Одной из распространенных причин является недостаточное обучение и ресурсы для обработки и хранения товаров. Если сотрудники не обучены должным образом обращаться с товарами и хранить их, это может привести к случайному повреждению. Кроме того, отсутствие надлежащего оборудования и ресурсов, таких как домкраты для поддонов и защитных упаковочных материалов, также может способствовать повреждению во время хранения и погрузочно-разгрузочных работ. Наконец, внешние факторы, такие как погода и стихийные бедствия, также могут привести к повреждению товаров на складе, если объект не подготовлен и не защищен должным образом. Все эти факторы могут привести к серьезным проблемам с контролем ущерба при управлении складом.

Склады часто сталкиваются с проблемой неточной инвентаризации, которая может иметь серьезные последствия для эффективности и прибыльности бизнеса. Это может произойти по разным причинам, например, из-за человеческой ошибки при подсчете или регистрации запасов, повреждения или утери предметов или несоответствий между физическими запасами и записями. Чтобы решить эту проблему, склады могут принять такие меры, как регулярная инвентаризация запасов, использование сканеров штрих-кодов и программного обеспечения для управления запасами, а также тщательное обучение сотрудников правильному обращению с запасами и ведению учета. Принимая меры по повышению точности, а также регулярно проверяя и обновляя учетные записи, склады могут снизить риск возникновения дефицита или избытка запасов и обеспечить бесперебойную работу своих операций.

Склады часто сталкиваются с проблемой избыточной деятельности, которая относится к задачам, которые не нужны и не добавляют ценности организации. Эти действия могут включать повторяющиеся ручные задачи, работу с документами и ввод данных, которые можно автоматизировать или исключить. Избыточные действия могут привести к неэффективности, ошибкам и увеличению затрат на склад. Это связано с тем, что на большинстве складов установлены устаревшие или неэффективные системы, требующие ненужных действий для выполнения задач. Плохо спроектированные рабочие процессы и отсутствие автоматизации также могут способствовать дублированию деятельности.

Неоптимальная комплектация на складах означает ситуацию, когда работники не могут подобрать необходимые товары наиболее эффективным способом. Это может привести к потере большого количества времени и ресурсов, а также к снижению производительности. Одна из основных причин, по которой склады сталкиваются с неоптимальной комплектацией, связана с отсутствием надлежащих систем и технологий для оптимизации процесса комплектации. Другой причиной может быть отсутствие обучения и четкого

общения между работниками, что приводит к путанице и ошибкам. Плохая планировка и дизайн склада также могут способствовать неоптимальному комплектованию, поскольку он может быть не оптимизирован для эффективного перемещения и комплектования товаров. В целом, устранение и улучшение неоптимального комплектования складов имеет решающее значение для успеха и эффективности бизнеса.

После реализации информационной системы (модель TO-VE) контекстная диаграмма верхнего уровня выглядит следующим образом (рисунок 11).



Рисунок 11 – Контекстная диаграмма функциональной модели системы в нотации IDEF0 модели «TO-VE»

Контекстная диаграмма верхнего уровня модели TO-VE разделена на четыре процесса, как показано на рисунке 12, образуя первый уровень модели TO-VE.

Бизнес-процессы, представленные в рамках данной диаграммы: приемка товара (рисунок 13), хранение товара (рисунок 14), комплектование товара по

требованию (рисунок 15), отгрузка товара (рисунок 16), проведение инвентаризации (рисунок 17).

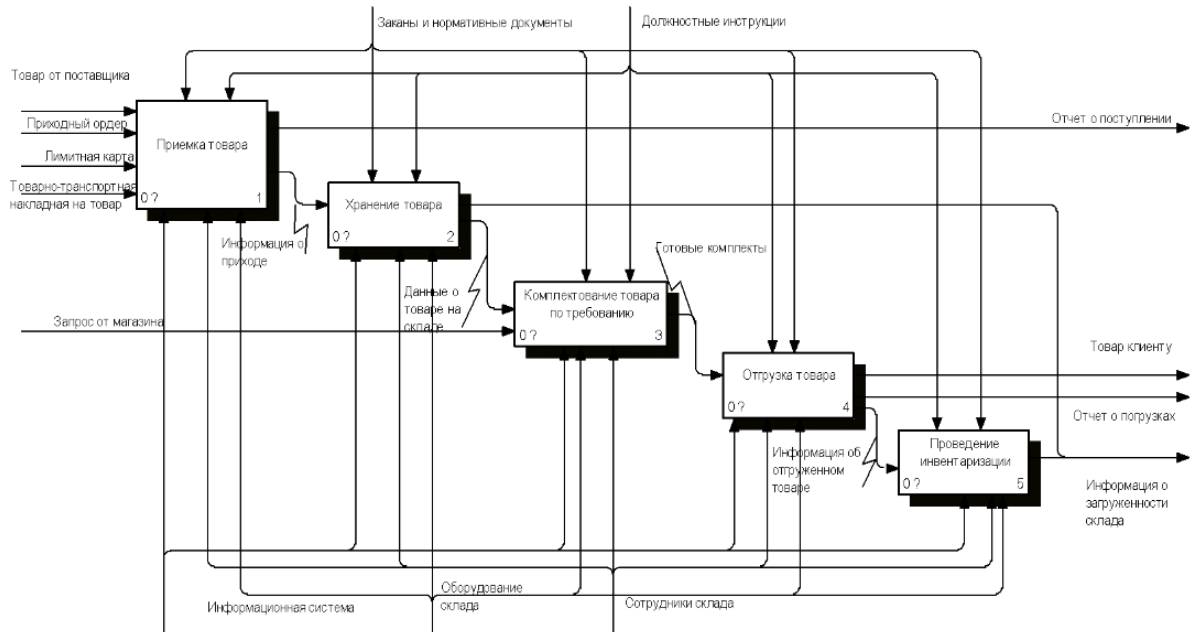


Рисунок 12 – Первый уровень функциональной модели системы в нотации IDEF0 модели «ТО-ВЕ»

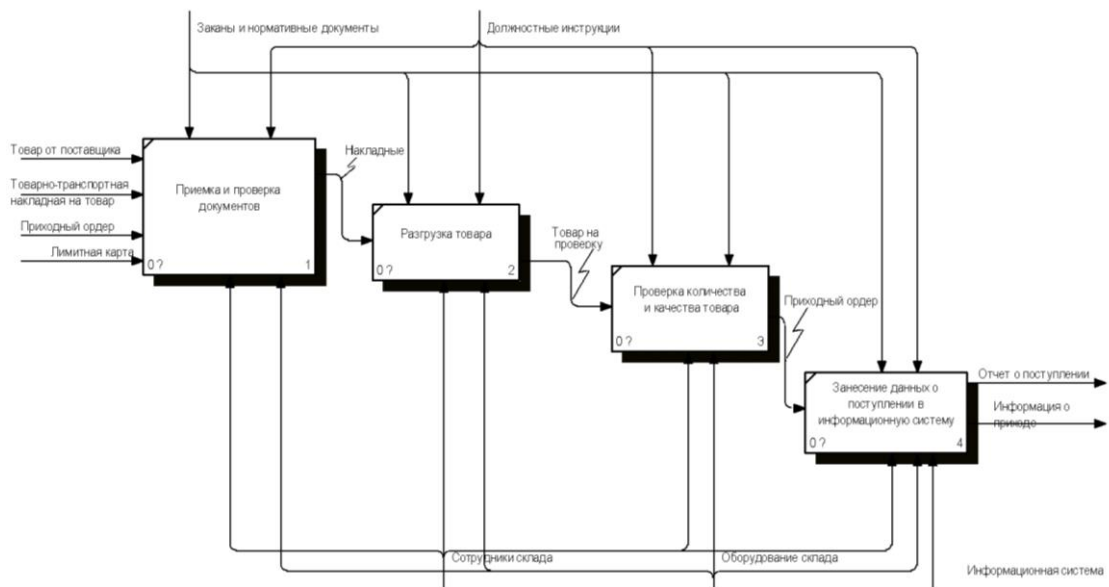


Рисунок 13 – Второй уровень функциональной модели системы в нотации IDEF0 модели «ТО-ВЕ»

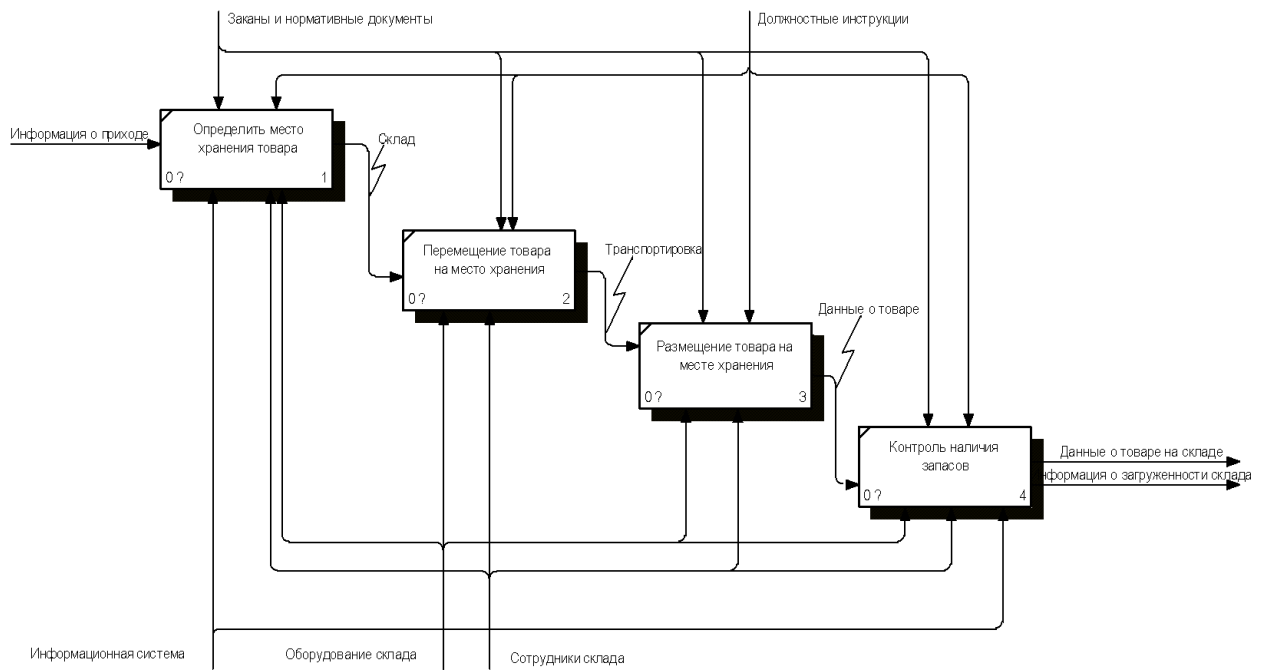


Рисунок 14 – Второй уровень функциональной модели системы в нотации IDEF0 модели «ТО-ВЕ»

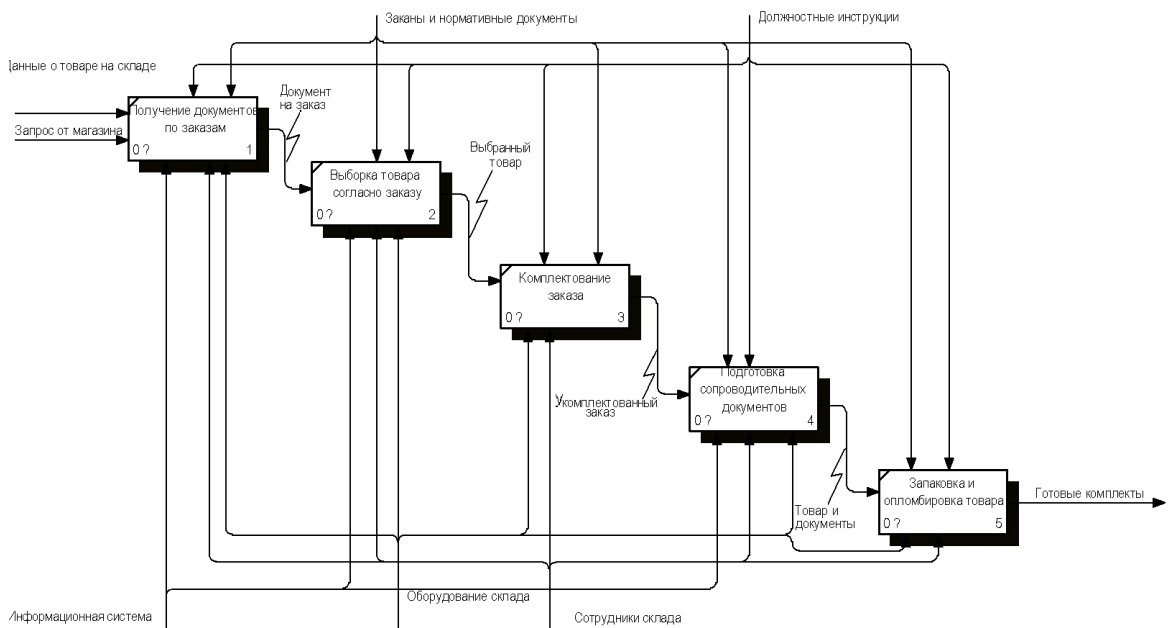


Рисунок 15 – Второй уровень функциональной модели системы в нотации IDEF0 модели «ТО-ВЕ»

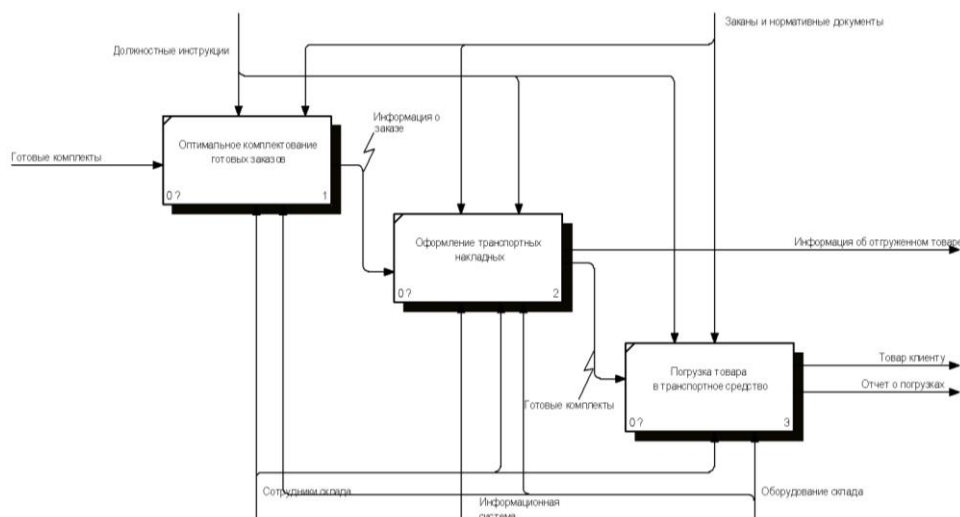


Рисунок 16 – Второй уровень функциональной модели системы в нотации IDEF0 модели «ТО-ВЕ»

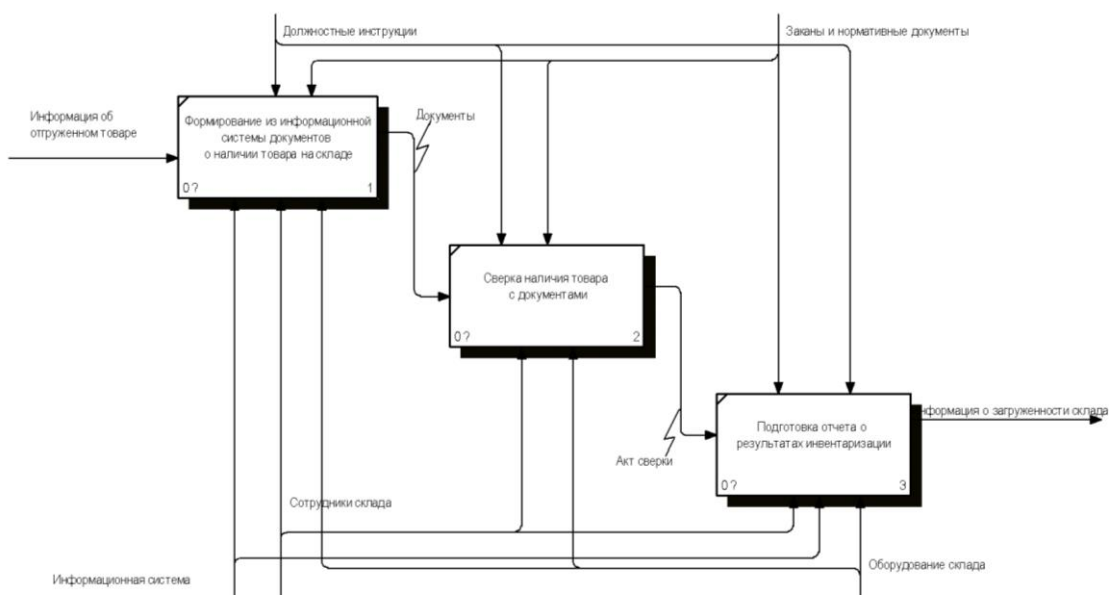


Рисунок 17 – Второй уровень функциональной модели системы в нотации IDEF0 модели «ТО-ВЕ»

Построенные функциональные модели помогли визуализировать бизнес-процесс и позволили приступить к проектированию системы автоматизации.

## 3.2 Проектирование структур хранения данных

Реляционная модель сущностей — это модель для идентификации сущностей, которые будут представлены в базе данных, и представления того, как эти сущности связаны.

На рисунке 18 представлена ER-диаграмма проектируемой БД.

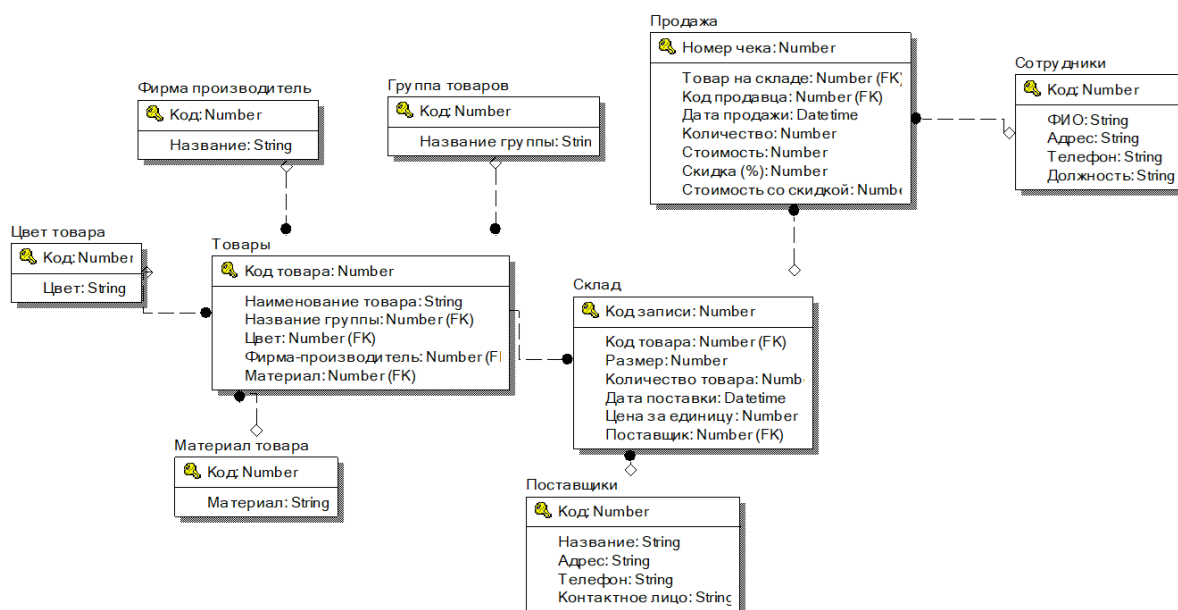


Рисунок 18 - ER-диаграмма разрабатываемой БД

Модель данных ER определяет корпоративную схему, которая графически представляет общую логическую структуру базы данных [16]. Диаграмма отношений сущностей объясняет отношения между сущностями, присутствующими в базе данных. ER-модели используются для моделирования объектов реального мира, таких как человек, автомобиль или компания, а также связей между этими объектами реального мира. Проще говоря, ER-диаграмма — это структурный формат базы данных [2].

В таблицах 1-9 показаны атрибуты таблиц.



Таблица 1 - Схема отношения Группа товаров

№ строки	Содержание поля	Имя поля	Тип, длина	Примечания
1	Код группы товаров	Код	INT(4)	Первичный ключ, уникальный
2	Название группы товаров	Название группы	CHAR(50)	Обязательное поле

Таблица 2 - Схема отношения Материал товара

№ строки	Содержание поля	Имя поля	Тип, длина	Примечания
1	Код материала товара	Код	INT(4)	Первичный ключ, уникальный
2	Название материала товара	Материал	CHAR(50)	Обязательное поле

Таблица 3 - Схема отношения Поставщики

№ строки	Содержание поля	Имя поля	Тип, длина	Примечания
1	Код фирмы-поставщика	Код	INT(4)	Первичный ключ, уникальный
2	Название фирмы-поставщика	Название	CHAR(50)	Обязательное поле
3	Адрес фирмы-поставщика	Адрес	CHAR(100)	
4	Телефон фирмы-поставщика	Телефон	INT(40)	Обязательное поле
5	ФИО человека для связи с фирмой поставщиком	Контактное лицо	CHAR(100)	

Таблица 4 - Схема отношения Продажа

№ строки	Содержание поля	Имя поля	Тип, длина	Примечания
1	Код чека при продаже	Номер чека	INT(4)	Первичный ключ, уникальный
2	Есть ли товар на складе	Товар на складе	INT(4)	Внешний ключ к Склад
3	Код продавца, совершившего продажу	Код продавца	INT(4)	Внешний ключ к Сотрудники
4	Дата осуществления продажи	Дата продажи	DATETIME	Обязательное поле

Продолжение таблицы 4

№ строки	Содержание поля	Имя поля	Тип, длина	Примечания
5	Количество проданных товаров	Количество	INT(4)	Обязательное поле
6	Стоимость продаваемого товара	Стоимость	INT(20)	Обязательное поле
7	Процент скидки	Скидка (%)	INT(20)	Обязательное поле
8	Итоговая сумма	Стоимость со скидкой	INT(20)	Обязательное поле

Таблица 5 - Схема отношения Склад

№ строки	Содержание поля	Имя поля	Тип, длина	Примечания
1	Код записи на складе	Код записи	INT(4)	Первичный ключ, уникальный
2	Код товара	Код товара	INT(1)	Внешний ключ к Товары
3	Размер завозимого товара	Размер	INT(4)	Обязательное поле
4	Количество завозимого товара	Количество товара	INT(4)	Обязательное поле
5	Стоимость одной единицы товара	Цена за единицу	INT(4)	Обязательное поле
6	Код фирмы поставщика	Поставщик	INT(1)	Внешний ключ к Поставщики

Таблица 6 - Схема отношения Сотрудники

№ строки	Содержание поля	Имя поля	Тип, длина	Примечания
1	Код сотрудника	Код	INT(4)	Первичный ключ, уникальный
2	ФИО сотрудника	ФИО	CHAR(50)	Обязательное поле
3	Адрес сотрудника	Адрес	CHAR(100)	
4	Телефон сотрудника	Телефон	INT(40)	Обязательное поле
5	Должность сотрудника	Должность	CHAR(100)	Обязательное поле

Таблица 7 - Схема отношения Товары

№ строки	Содержание поля	Имя поля	Тип, длина	Примечания
1	Код товара	Код товара	INT(4)	Первичный ключ, уникальный
2	Наименование товара от фирмы поставщика	Наименование товара	CHAR(50)	Обязательное поле
3	Код группы товаров, к которой он относится	Название группы	INT(1)	Внешний ключ к Группа товаров
4	Код цвета товара	Цвет	INT(1)	Внешний ключ к Цвет товара
5	Код фирмы производитель	Фирма-производитель	INT(1)	Внешний ключ к Фирма производитель
6	Код материала товара	Материал	INT(1)	Внешний ключ к Материал товара

Таблица 8 - Схема отношения Фирма производитель

№ строки	Содержание поля	Имя поля	Тип, длина	Примечания
1	Код фирмы производителя	Код	INT(4)	Первичный ключ, уникальный
2	Название фирмы производителя	Название	CHAR(50)	Обязательное поле

Таблица 9 - Схема отношения Цвет товара

№ строки	Содержание поля	Имя поля	Тип, длина	Примечания
1	Код цвета товара	Код	INT(4)	Первичный ключ, уникальный
2	Название цвета товара	Цвет	CHAR(50)	Обязательное поле

В приведенных выше таблицах атрибуты позволяют описать свойства и характеристики информации, которая будет использоваться в базах данных. Это дает возможность оптимально перенести бизнес-процессы в правила хранения данных, их ввод в базу, создание связей между таблицами и обеспечения целостности.

### 3.3 Описание реализации вариантов использования

Вариант использования описывает, как система или приложение ведет себя при определенных условиях. Чем более детально будет описан сценарий действий, тем точнее получится выстроить варианты поведения. Целью определения вариантов использования является достижение согласия между участниками проекта относительно поведения и области действия системы [15].

Варианты использования описывают поведение системы, особенно взаимодействие между системой и ее пользователями. Пользователь — это человек, роль, организация или другая система. Являясь актором он взаимодействует с системой для достижения определенной цели в определенной последовательности действий. Название варианта использования обычно вытекает из цели (например, «контролировать скорость транспортного средства» или «выводить деньги»), чтобы с первого взгляда было понятно, какое поведение системы описывается [12].

Варианты использования – немецкий перевод варианта использования – используются при разработке программного обеспечения, создании систем и продуктов с начала 1990 года. Они также очень популярны сегодня, поскольку документируют функциональность существующей или планируемой системы с помощью простых моделей. Из плюсов:

- это значительно увеличивает общее понимание взаимодействия между актором и системой;
- сценарии могут быть идентифицированы;
- функциональные требования и соответствующие тестовые примеры могут быть получены из целей участников.

Диаграммы вариантов использования визуализируют варианты использования и участников с соответствующими отношениями. Они дают хороший обзор всей системы, но, в отличие от спецификаций, описывают не процессы, а скорее связи между рядом вариантов использования и

участвующими сторонами. Они считаются графическим представлением вариантов использования, а унифицированный язык моделирования (UML) определяет их как так называемую диаграмму поведения. Наиболее важными элементами модели являются: актеры, случаи использования, отношения, границы системы. Спецификации и диаграммы можно использовать вместе, чтобы обеспечить дополнительное представление о поведении системы. Это позволит получить более подробную схему и выявить сильные и слабые стороны.

Варианты использования — хороший способ описать и понять систему с точки зрения пользователя. Первой проблемой для компаний при его использовании является поиск тем и контента. Чем больше сотрудников задействовано, тем сложнее становится этот этап. При этом результаты и вытекающие из них требования становятся более детализированными. При формулировании организациям следует проявлять осторожность и создавать не слишком сложные подробные описания. Здесь требуется некоторая практика.

Использование стандартных текстов при описании триггеров, пред- и постусловий не рекомендуется. Поскольку шаблонные тексты не смогут тонко сформулировать требуемое. Последовательности реализации часто могут быть получены из предусловий и постусловий: вариант использования с постусловиями X может быть реализован раньше другого с предусловием X.

Исходя из анализа технического задания в системе будут следующие актеры:

- бухгалтер – осуществляет проверку работы склада и верное заполнение документации и отчетности;
- сотрудник склада – отвечает за работу склада;
- сотрудник магазина – делает оформление заказа.

Исходя из потребностей действующих лиц, можно выделить следующие ключевые варианты использования, описывающие необходимые действия на торговом предприятии: внутрискладское перемещение товара, заказ товара со

склада, инвентаризация, комплектация заказов для отгрузки, контроль работы сотрудников, определение места хранения товара, подготовка документов для транспортировки, прием товара со склада, приемка поступившей продукции, проверка документации, составление отчетов.

Далее приведем диаграмму вариантов использования на рисунке 19, на котором визуализируем выявленные варианты.

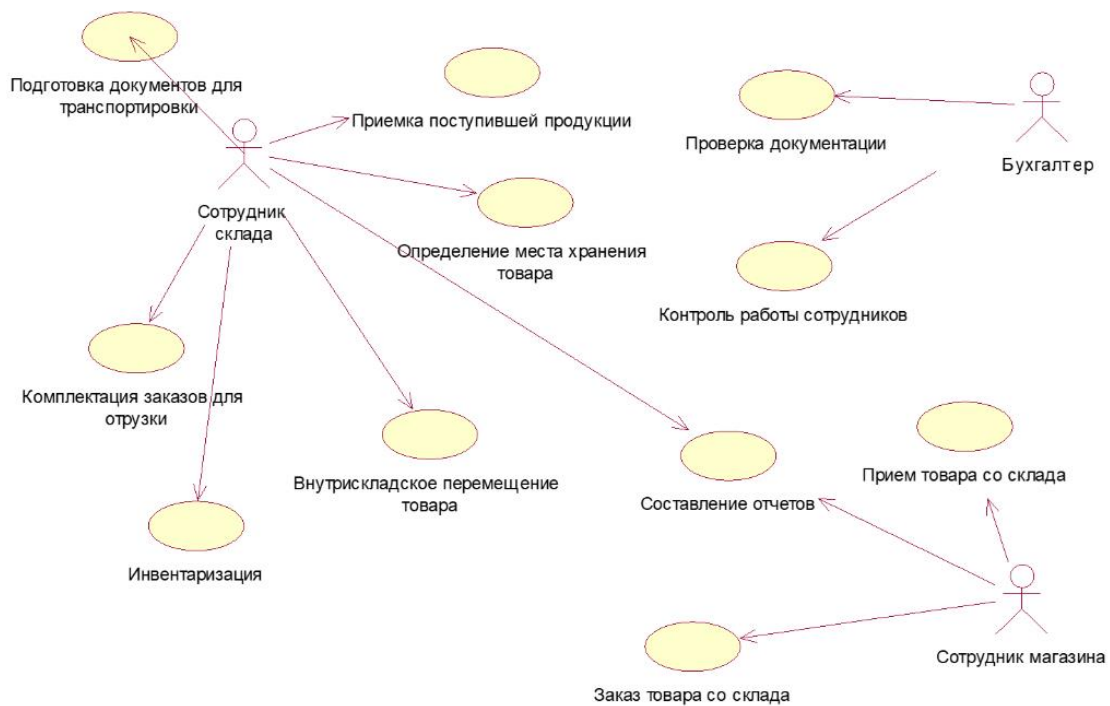


Рисунок 19 – Диаграмма вариантов использования системы

Во время реализации разбиение вариантов использования на фрагменты, которые реализуются в рамках спринта, оказывается очень полезным. На практике использование случаев неправильного использования также активно способствует удовлетворению потребностей клиентов, тем более что их также можно визуализировать с помощью диаграммы. Благодаря визуализации производители могут лучше поставить себя на место своих пользователей и определить конкретные требования [6].

## Глава 4 Реализация программной системы

### 4.1 Разработка алгоритмов реализации вариантов использования

В UML диаграмма деятельности дает представление о поведении системы, описывая последовательность действий в процессе. Диаграммы действий имеют сходство с блок-схемами, поскольку они представляют поток между действиями в деятельности. Однако параллельные или одновременные потоки, а также альтернативные потоки также могут быть представлены на диаграммах деятельности. Благодаря данному инструменту моделирования и визуализации разработчики понимают задачи и требования к системе. UML диаграмма дает обширную возможность коммуникации разработчиков между собой, взаимодействию с заказчиком и сопоставлению требований с реализацией.

Диаграммы действий UML относятся к группе диаграмм поведения в UML. В то время как структурная диаграмма фиксирует состояние системы, то есть существующие объекты и их иерархии, а также связи друг с другом в определенный момент времени, диаграммы поведения описывают хронологический поток потоков данных. Помимо диаграммы деятельности к этой группе относятся диаграмма вариантов использования и диаграмма конечного автомата. Диаграммы действий по использованию и обозначениям аналогичны блок-схемам (особенно блок-схемам программ), но адаптированы для объектно-ориентированного программирования [4].

Ниже, на рисунке 20 представлена диаграмма активности для склада. Выстраивая взаимодействия между ключевыми сотрудниками, происходит оптимизация действий и исключаются дополнительные лишние этапы.

Диаграмма состояний, также называемая диаграммой состояний, диаграммой конечного автомата или диаграммой перехода состояний, визуализирует последовательность состояний, которые объект может принять в своем жизненном цикле. Он используется для описания поведения системы,

подсистемы, компонента или класса. Использование интерфейсов в системе также можно указать с помощью диаграмм состояний. Особое внимание уделяется переходам между различными состояниями объекта, триггерным действиям и свойствам объекта, которые он имеет или должен иметь до изменения состояния [10].

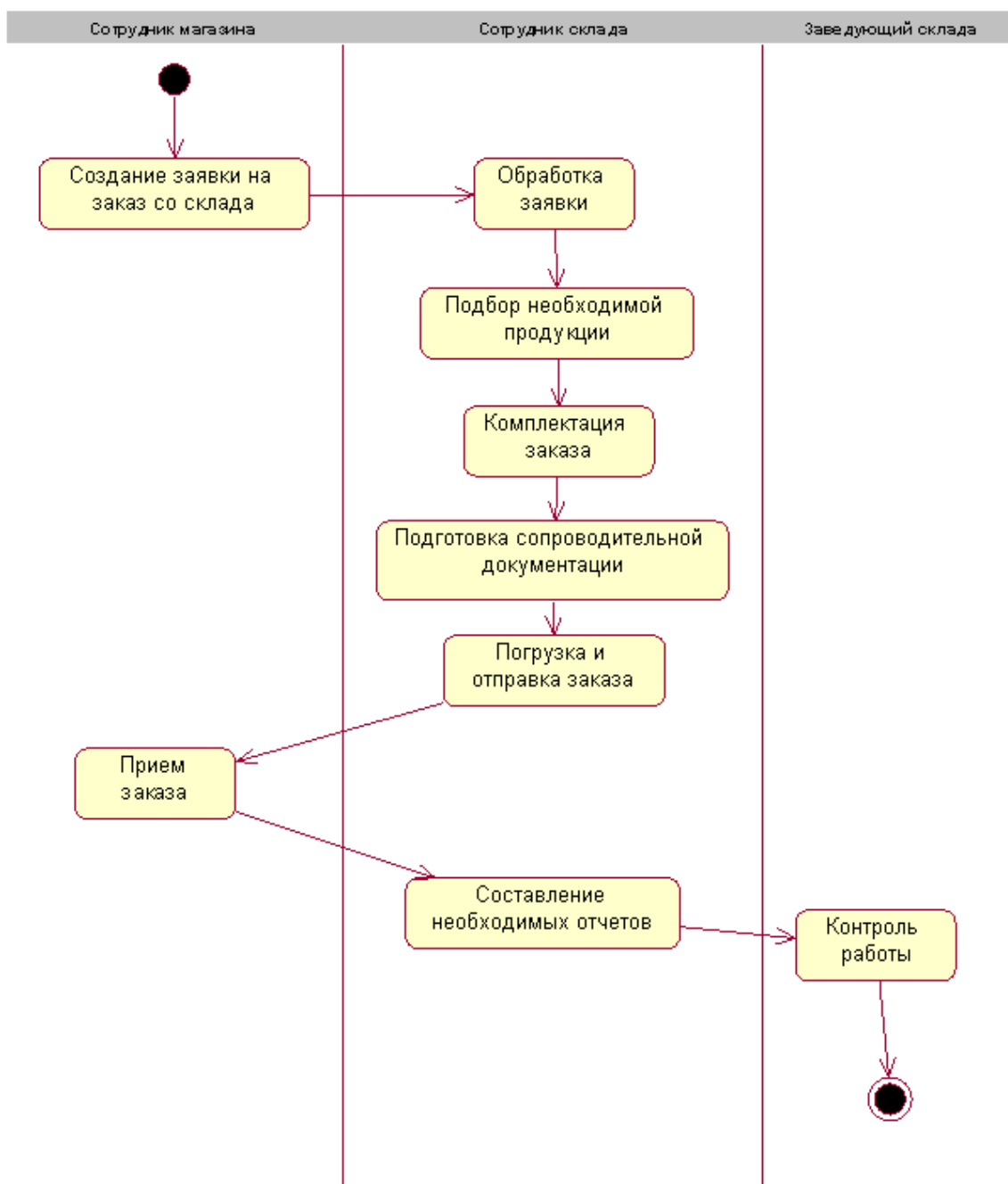


Рисунок 20 – Диаграмма активности



Диаграммы состояний основаны на концепции, разработанной Дэвидом Харелем, израильским профессором информатики, опубликовавшим книгу «Диаграммы состояний: визуальный формализм для сложных систем» в 1987 году. Лишь позже диаграмма состояний была включена как один из 14 типов диаграмм в унифицированный язык моделирования (UML) и, соответственно, в язык моделирования систем (SysML). Концепция была расширена и теперь включает иерархически вложенные состояния, ортогональные области и действия, которые могут одновременно зависеть от состояния системы и запускающего события. Также было интегрировано использование действий входа и выхода, которые больше ориентированы на состояние, чем на действие.

На рисунке 21 представлена диаграмма состояний для состояния классов для работы на складе.

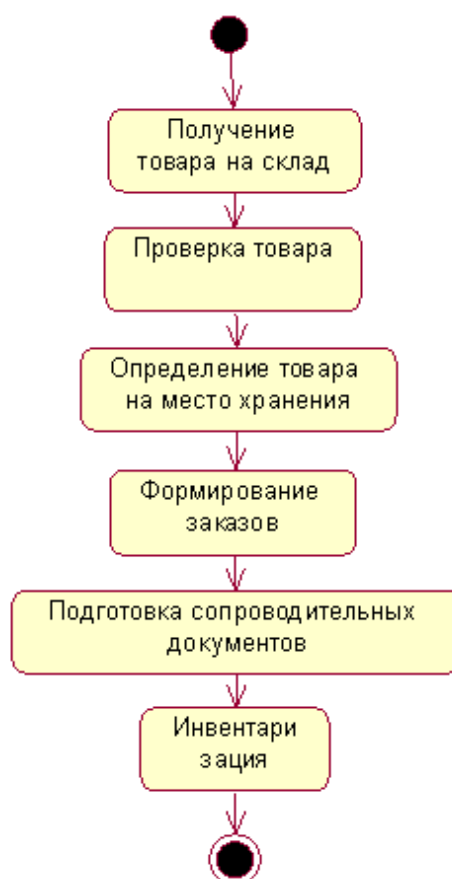


Рисунок 21– Диаграмма состояний

Диаграммы состояний часто используются в области встроенных систем для мониторинга, управления и регулирования функций или обработки данных или сигналов. Встроенные системы – сочетание программного и аппаратного обеспечения – часто выполняют невидимые для пользователя услуги в автомобилях, самолетах, холодильниках или лифтах. Они контролируют выдачу наличных в банкомате и обеспечивают Aqua Стоп у посудомоечной машины.

Целые секторы экономики, такие как медицинские технологии, индустрия развлечений, индустрия связи и транспорт, также полагаются на встроенные системы и, следовательно, на диаграммы состояний. А использование диаграмм состояний широко распространено даже в различных областях компании, которые занимаются производством продуктов в самом широком смысле, например, управление продуктами и проектами или разработка требований.

Таким образом разработка алгоритмов реализации является важным этапом разработки автоматизированной системы. Благодаря этому можно наглядно выстроить последовательность действий в системе, которые позволят достичь главную цель разработки – автоматизацию процессов. Благодаря алгоритмам вариантов использования определено, какие действия должна выполнять программа в ответ на действия пользователя.

## **4.2 Разработка классов информационной системы**

Верхняя часть диаграммы классов — это имя класса, которое представляет имя класса. Средняя часть включает атрибуты класса, то есть переменные и методы класса. Нижняя часть состоит из операций класса, т.е. действий, которые можно выполнить с классом [14].

На рисунке 22 представлена диаграмма классов для склада.

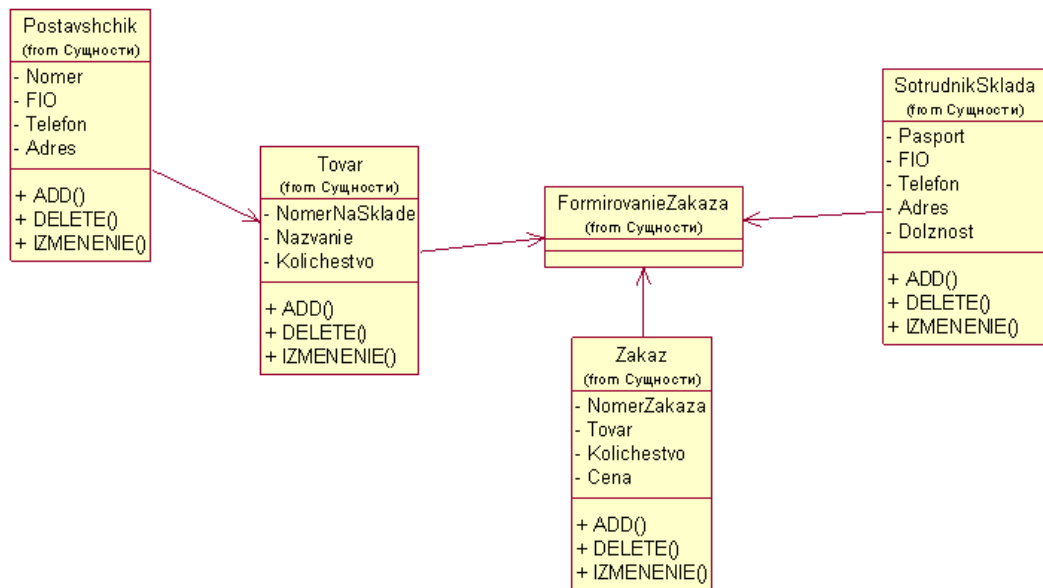


Рисунок 22 – Диаграмма классов

На рисунке 23 представлена диаграмма «сущность-связь» для склада.

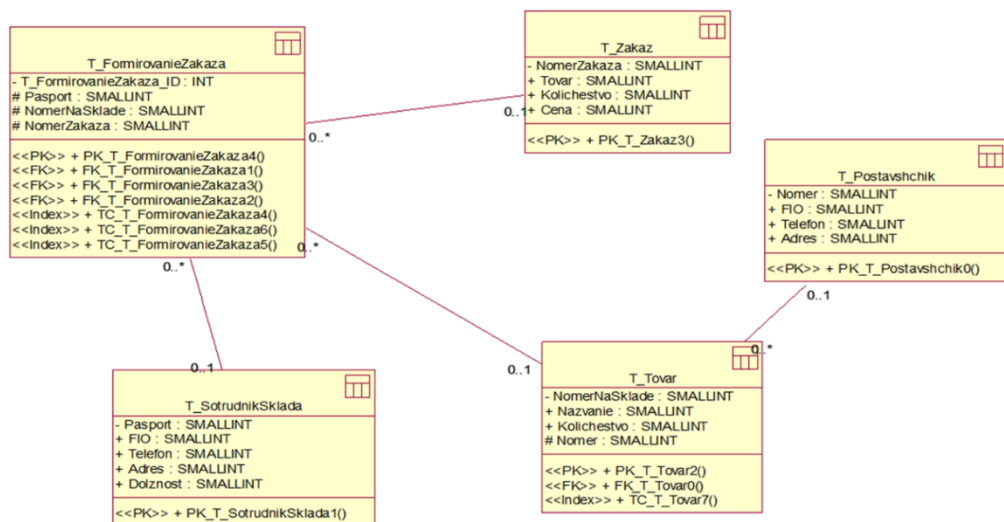


Рисунок 23 – Диаграмма «сущность-связь»

Диаграммы классов — часто используемый инструмент для представления структуры классов и их отношений. Они могут помочь понять,

как работает программа, и могут быть полезны для устранения неполадок или генерации кода [11].

### 4.3 Разработка графического интерфейса

Windows Forms (WinForms) как часть Microsoft .NET Framework позволяет легко и быстро создавать графические пользовательские интерфейсы (GUI) для настольных приложений Windows [9].

Вид главной страницы информационной системы проиллюстрирован на рисунке 24.

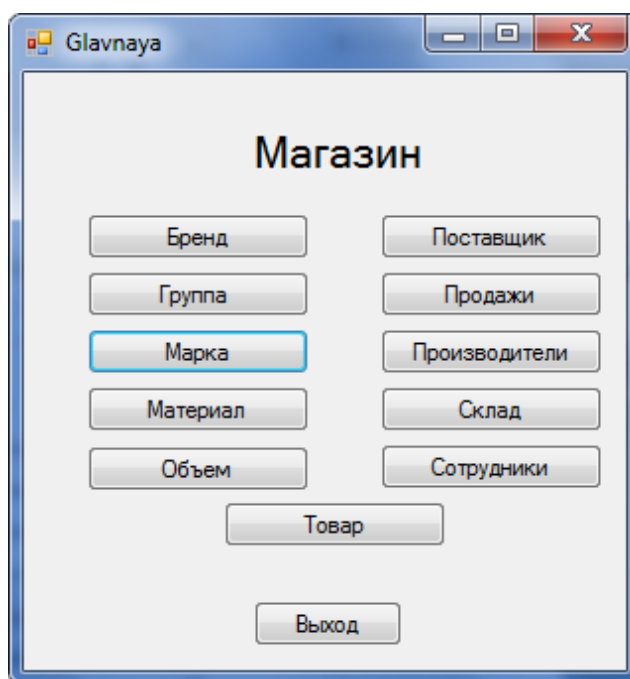


Рисунок 24 – Вид главной страницы информационной системы

Ниже (рисунки 25-35) проиллюстрирован внешний вид всех подчиненных страниц.

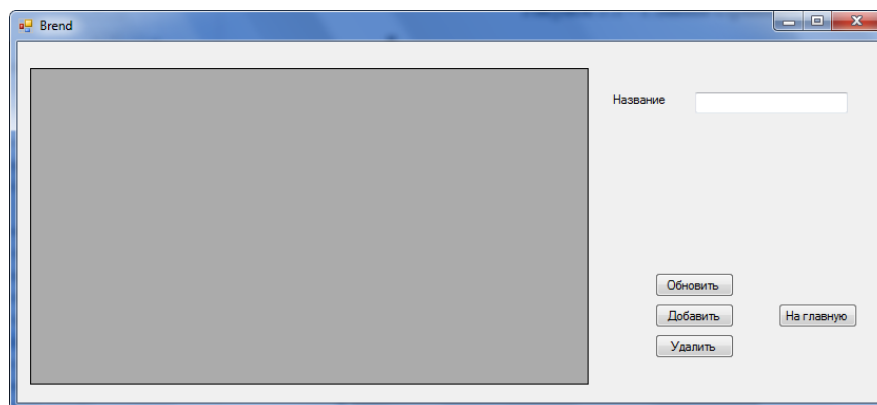


Рисунок 25 – Вид подчиненной страницы «Бренд» информационной системы

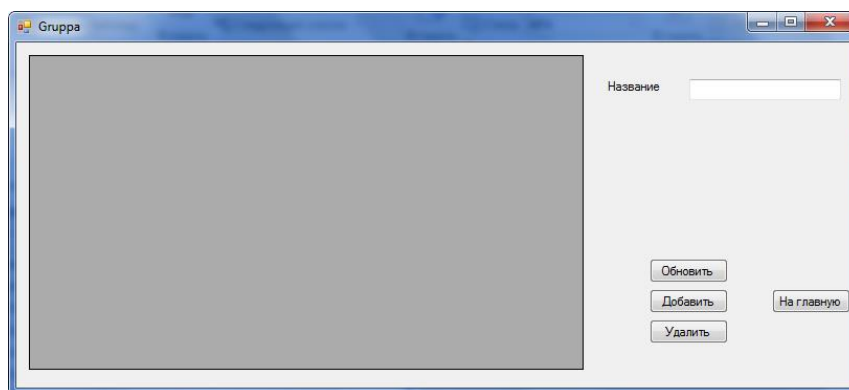


Рисунок 26 – Вид подчиненной страницы «Группа» информационной системы

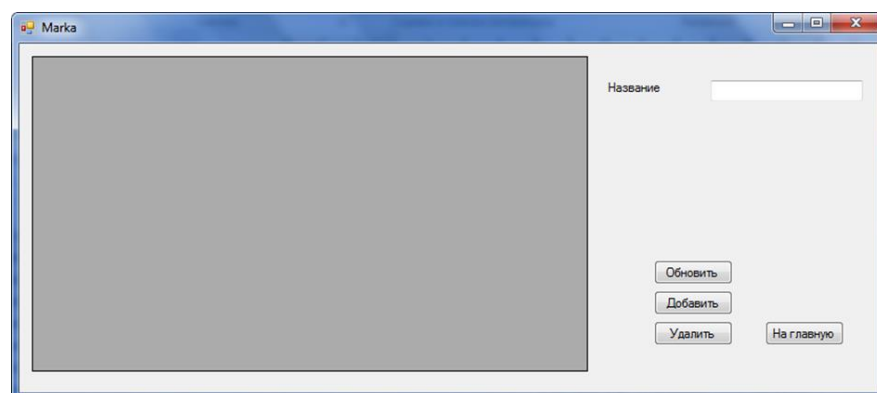


Рисунок 27 – Вид подчиненной страницы «Марка» информационной системы

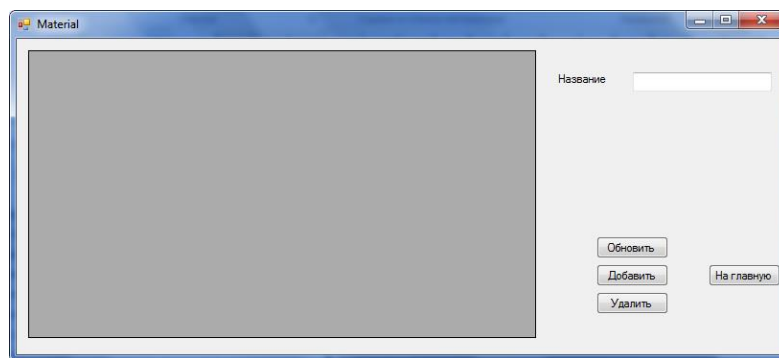


Рисунок 28 – Вид подчиненной страницы «Материал»  
информационной системы

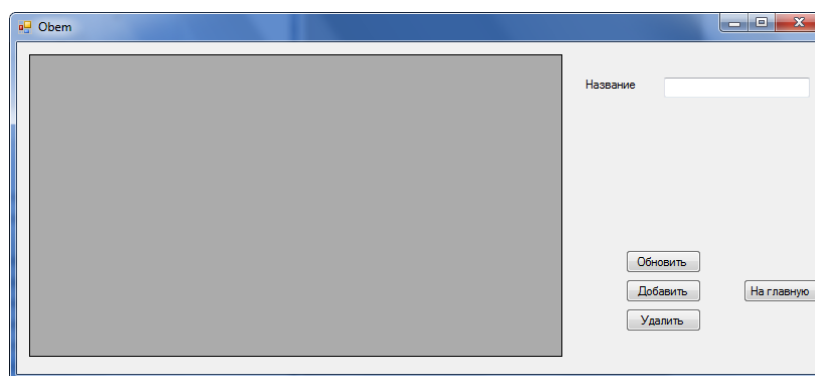


Рисунок 29 – Вид подчиненной страницы «Объем» информационной  
системы

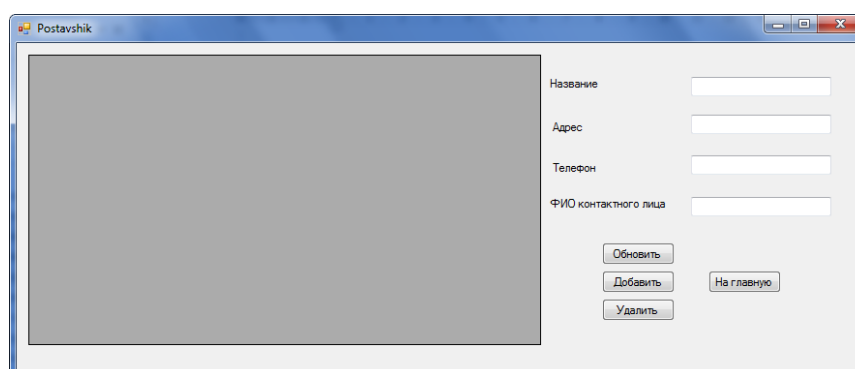


Рисунок 30 – Вид подчиненной страницы «Поставщики»  
информационной системы

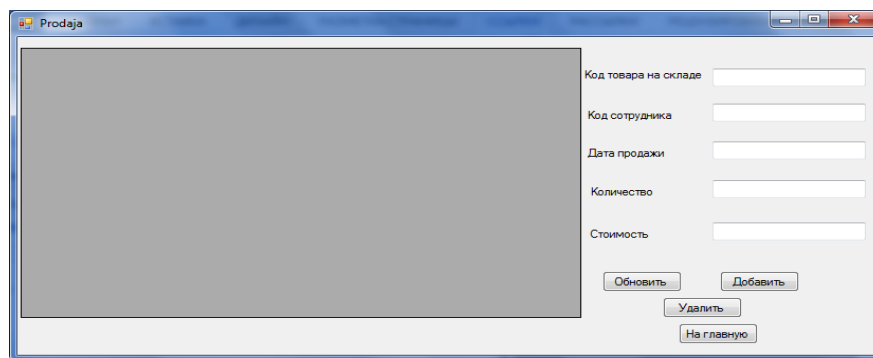


Рисунок 31 – Вид подчиненной страницы «Продажи» информационной системы

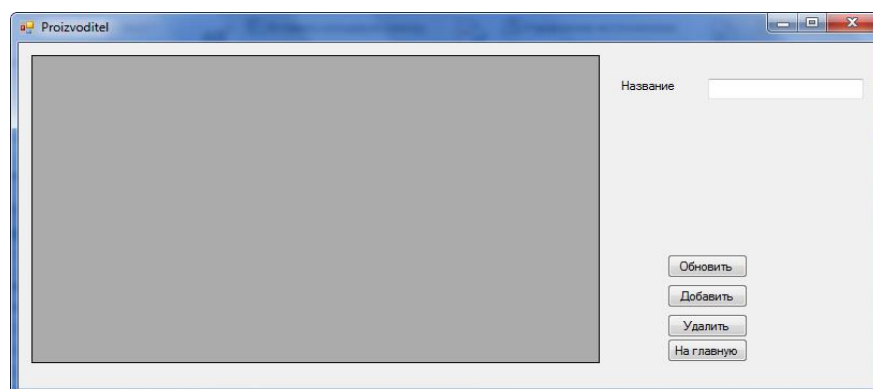


Рисунок 32 – Вид подчиненной страницы «Производители» информационной системы

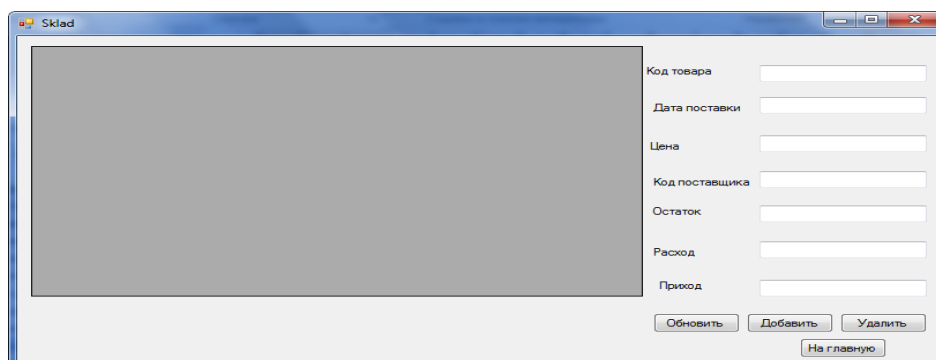


Рисунок 33 – Вид подчиненной страницы «Склад» информационной системы

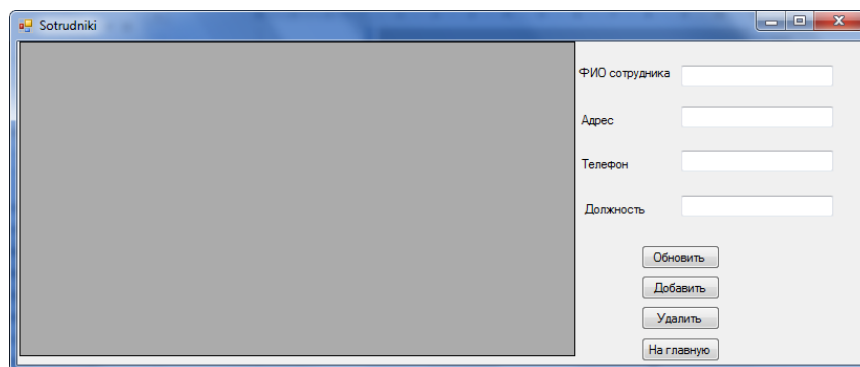


Рисунок 34 – Вид подчиненной страницы «Сотрудники» информационной системы

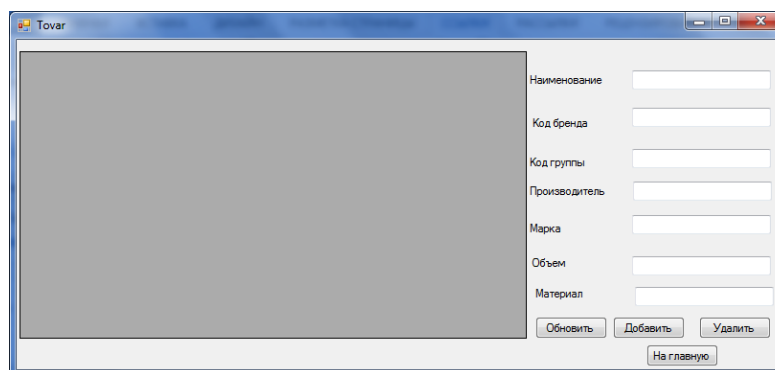


Рисунок 35 – Вид подчиненной страницы «Товар» информационной системы

На странице можно просматривать данные, удалять строки таблицы и добавлять информацию. Чтобы увидеть изменения, необходимо нажать кнопку «Обновить».

#### 4.4 Тестирование программной системы

Тестирование «черного ящика» — это метод тестирования программного обеспечения, аналогичный тестированию «белого ящика». Черный ящик также известен как поведенческое тестирование/функциональное тестирование/тестирование закрытого ящика.



Это метод анализа функциональных и нефункциональных аспектов программного обеспечения. В центре внимания тестировщика программного обеспечения не лежит внутренняя структура/реализация и дизайн задействованной системы. Это означает, что тестировщику программного обеспечения не обязательно владеть знаниями в области программирования и не обязательно иметь доступ к коду. Тестирование «черного ящика» было разработано таким образом, чтобы его можно было применять для анализа требований клиентов и анализа спецификаций. Во время тестирования программного обеспечения «черного ящика» тестер программного обеспечения выбирает набор допустимых и недействительных входных данных и ищет действительный выходной ответ.

Тестирование черного ящика — это метод, который не учитывает внутреннюю структуру, дизайн и реализацию продукта, подлежащего тестированию. Другими словами, тестер не знает его внутреннего функционирования. Черный ящик оценивает только внешнее поведение системы. Проверяются входные данные, полученные системой, а также выходные данные или ответы, которые она производит.

Метод тестирования белого ящика — это метод, который рассматривает код и структуру тестируемого продукта и использует эти знания для выполнения тестов. Этот метод используется на этапе модульного тестирования, хотя он также может применяться и на других этапах, таких как интеграционные тесты. Для реализации этого метода тестировщик или человек, который будет использовать этот метод, должен обладать обширными знаниями технологии, используемой для разработки программы.

Комбинацией данных видов тестирования образовывается уникальный метод тестирования серого ящика. Для данного метода необходимо обладать частичными познаниями в разработке программных систем и внутренних процессов. Проведение тестирования данным методом позволяет улучшить покрытие кода. Тестовые случаи могут показать различные пути реализации программы. Но ключевым недостатком может являться ограничение доступа

к исходному коду, а значит на результативное построение тестовых ситуаций может быть затрачено больше времени [3].

Для тестирования системы добавим в таблицы через приложение несколько строк в таблицу «Продажи».

Откроем страницу Продажа в приложении (рисунок 36).

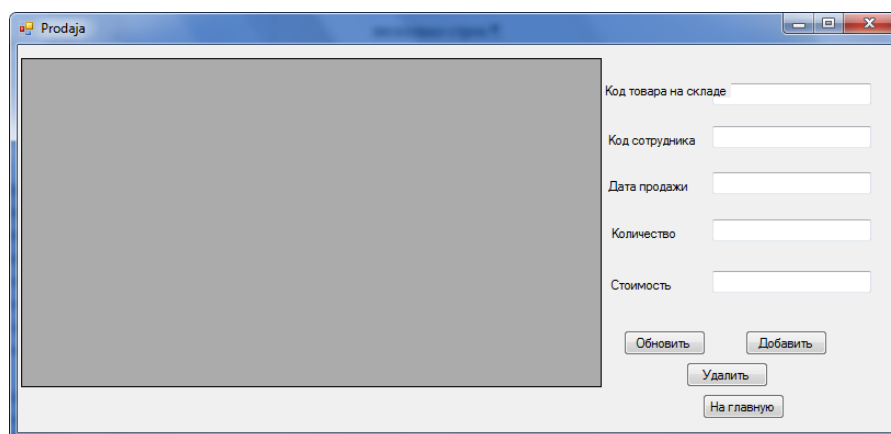


Рисунок 36 – Вид подчиненной страницы «Продажа» (тест)

После нажатия кнопки «Обновление» появится таблица, представленная на рисунке 37.

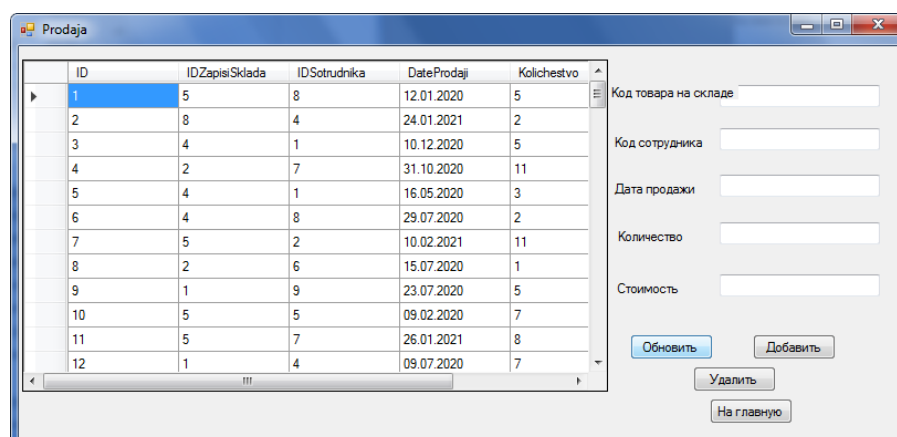


Рисунок 37 – Вид подчиненной страницы «Продажа» (тест)

Введем необходимые данные в поля (рисунок 38).

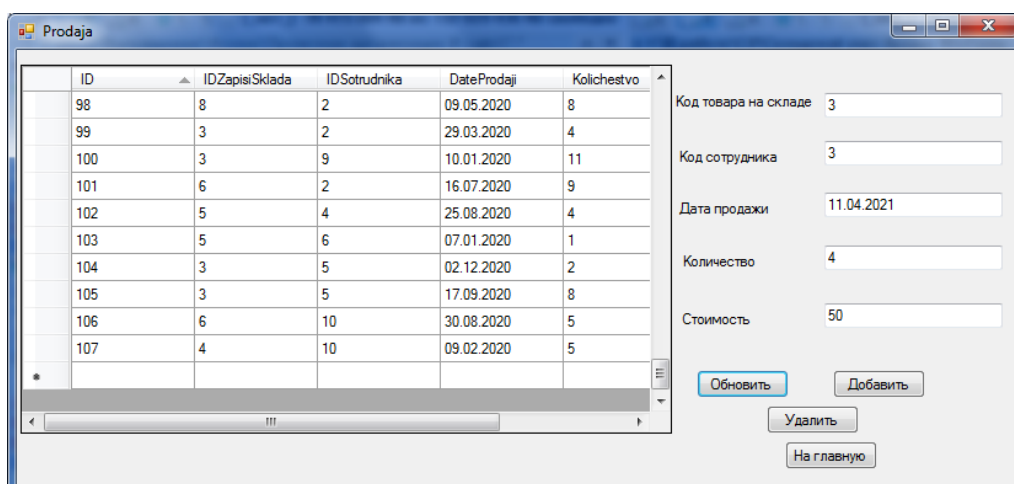


Рисунок 38 – Вид подчиненной страницы «Продажа» (тест)

При нажатии кнопки «Добавить» информация добавится в таблицу. Чтобы просмотреть изменение нажмем кнопку «Обновить». Результат показан на рисунке 39.

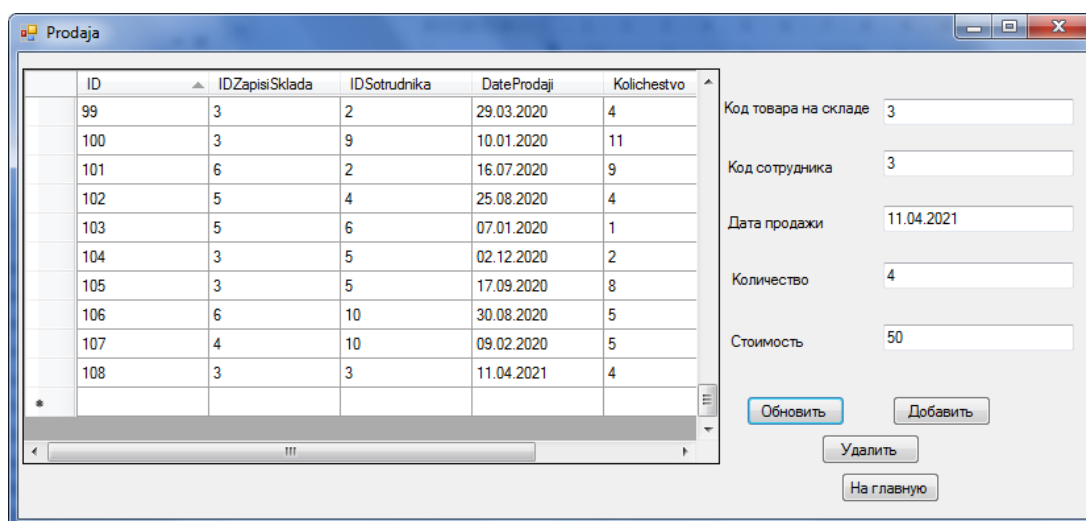


Рисунок 39 – Вид подчиненной страницы «Продажа» (тест)

Выделим добавленную строку и при нажатии кнопки «Удалить» данная запись удалится. После нажатии кнопки «Обновить» строка удалится (рисунок 40).

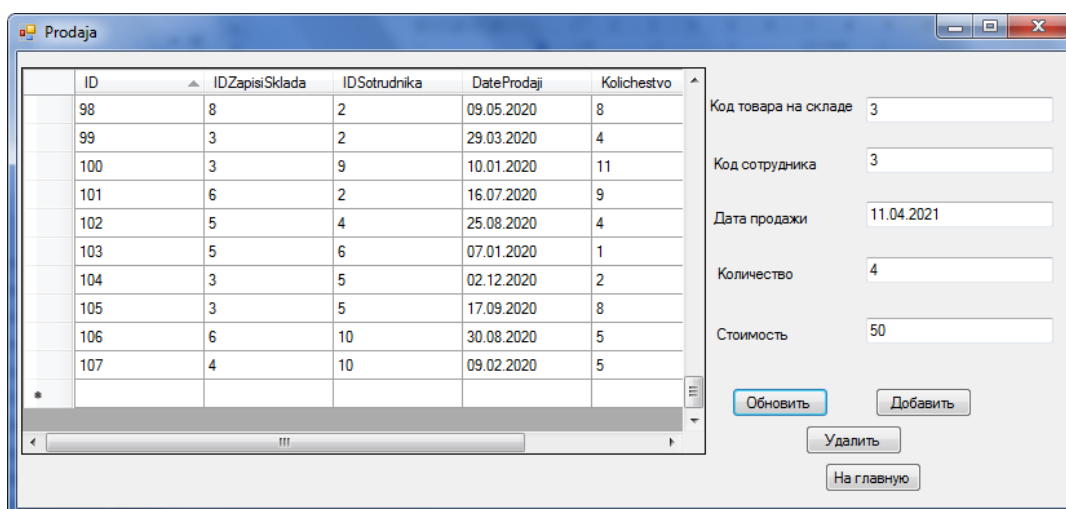


Рисунок 40 – Вид подчиненной страницы «Продажа» (тест)

Проведенное тестирование программного обеспечения не выявило ошибок во время исполнения.

Тестирование программного обеспечения является важным этапом в процессе разработки продукта. Оно помогает обнаружить ошибки, повысить качество разрабатываемого продукта, ускорить процесс внедрения реализации программного обеспечения и снизить возможные риски потери данных для пользователей программы. Кроме того, тестирование позволяет улучшить обратную связь с пользователями и повысить их удовлетворенность продуктом. В целом, проведение тестирования программного обеспечения приносит множество положительных результатов для всех участников процесса разработки.

#### 4.5 Экономическое обоснование проекта

Собственные затраты ( $S$ ) рассчитываются по данной формуле (1):

$$S = P + E_n * K, \quad (1)$$

где  $P$  - эксплуатационные расходы на функционирование РСОИ, у.е.;

$E_n$  - нормативный коэффициент приведения затрат к единому году;

$K$  - капитальные (единовременные) затраты на разработку РСОИ, у.е.

Для вычислительной техники  $E_n = 0.25...0.35$ .

Капитальные затраты на разработку и внедрение распределённой системы обработки информации рассчитываются по следующей формуле (2):

$$K = K_{pp} + K_{ob} + K_{po} + K_{pm} + K_{mnp}, \quad (2)$$

где  $K_{pp}$  - предпроизводственные затраты (проектирование РСОИ, научно-исследовательские, опытно-конструкторские работы), у.е.;

$K_{ob}$  - затраты на основное оборудование РСОИ, у.е.;

$K_{po}$  - затраты на программное обеспечение РСОИ, у.е.;

$K_{pm}$  - затраты на строительство и ремонт помещений для РСОИ, у.е.;

$K_{mnp}$  - единовременные затраты на монтаж, наладку и пуск РСОИ, у.е.

Затраты, связанные с проектированием РСОИ рассчитаны по формуле (3):

$$K_{pp} = \sum_{i=1}^N Z_i \cdot T f + D_{pr}, \quad (3)$$

где  $N$  - количество категорий сотрудников, привлечённых к разработке, чел.;

$Z_i$  - заработная плата работников  $i$  - категории, у.е. в месяц;

$T$  - время работы, месяцев;

$f$  - коэффициент отчислений на заработную плату;

$D_{pr}$  - прочие расходы (затраты на командировку, телефон и почтовые услуги), как правило, равные 0.7 стоимости оборудования [5].

Исходя из таблицы 10, следует:  $K_{pp} = 3040,56$  у.е.

Таблица 10 - Расчет заработной платы и отчислений

Статьи затрат	Сумма, у.е.
Средний оклад сотрудника	1854,00
Заработная плата с учетом надбавки (25%)	2317,50
Отчисления из заработной платы (39%)	723,06
Итого затраты на заработную плату (в месяц)	3040,56

Затраты на основное оборудование РСОИ рассчитано по формуле (4):

$$K_{ob} = C_{ob} \cdot (1 + k_{us}), \quad (4)$$

где  $C_{ob}$  - балансовая стоимость оборудования (в работе это смета на аппаратную часть проектируемой системы);

$k_{us} = 0.01...0.02$  - коэффициент установки.

Затраты на программное обеспечение -  $K_{po}$  можно определить прямым счётом.

Исходя из данных в таблице 11, следует:  $C_{ob} = 1565,89$  у.е.;  
 $K_{ob} = 1597,21$  у.е.;  $K_{po} = 496,85$  у.е.

Таблица 11 - Затраты на приобретаемое оборудование и обеспечение

№ п/п	Наименование оборудования и программ	Количество, шт	Цена за единицу, у.е.
1.	ПО	1	496,85
2.	Компьютер с комплектующими	1	1565,89

Затраты на монтаж, наладку и пуск РСОИ -  $K_{mnp}$ . Для предварительных расчётов затраты принимаются на уровне 10 - 20% стоимости основного оборудования  $C_{ob}$ : (формула 5).

$$K_{mnp} = (0.1...0.2) \cdot C_{ob} . \quad (5)$$

Следовательно:  $K_{mnp} = 313,18$  у.е.

Затраты на строительство и реконструкцию помещений  $K_{pm}$ : (формула б).

$$K_{pm} = (0.2...0.5) \cdot C_{ob} . \quad (6)$$

Следовательно:  $K_{pm} = 782,95$  у.е.

Общая величина капитальных (единовременных) затрат составляет 2.16...2.62 балансовой стоимости основного оборудования и равна:  $K = 5796,43$  у.е.

Эксплуатационные расходы характеризуют себестоимость обработки информации и складываются из трёх составляющих в формулу (7):

$$P = P_{osn} + P_{aup} + P_{obs}, \quad (7)$$

где  $P_{osn}$  - заработная плата основного производственного персонала, у.е.;

$P_{aup}$  - заработная плата административно - управленческого персонала (АУП), которая определяется как  $0,25 \cdot P_{osn}$ , у.е.;

$P_{obs}$  - общепроизводственные расходы, у.е.

Исходя из данных в таблице 12, следует:  $P_{osn} = 3490,00$  у.е..  
 $P_{aup} = 872,50$  у.е.

Таблица 12 - Затраты на оплату основного производственного персонала

№ п/п	Наименование основного производственного персонала	Количество, шт	Цена за единицу, у.е.
1.	Программист	1	1745
2.	Настройщик	1	1745

Общепроизводственные эксплуатационные расходы рассчитываются по следующей формуле (8) и включают в себя:

$$P_{obs} = P_{el} + P_{rm} + P_{rem} + P_{zap} + P_{ks} + P_{pr}, \quad (8)$$

где  $P_{ei}$  - расходы на электроэнергию в год, связанные с эксплуатацией ИТ-структуры, у.е.;

$P_{rm}$  - затраты на расходные материалы, связанные с функционирование ИТ-структуры, у.е.;

$P_{rem}$  - затраты, связанные с ремонтом оборудования (заработная плата сотрудников, осуществляющих ремонт, либо стоимость договора со сторонней организацией - ИТ аутсорсинг), у.е.;

$P_{zap}$  - затраты, связанные с приобретением запчастей для ремонта оборудования, у.е.;

$P_{ks}$  - затраты на аренду каналов связи (подключение Интернет), у.е.;

$P_{pr}$  - прочие расходы (чаще всего прочие расходы определяют, как  $P_{pr} = 0.6 \cdot P_{obs}$ , у.е.).

Исходя из таблицы 13 получим:  $P_{el} = 633,6$  у.е.

Таблица 13 - Затраты на потребление электроэнергии оборудованием РСОИ

№ п/п	Оборудование	Количество, шт.	Потребление, кВт*ч	Дней в году работает	Часов в сутки работает	Стоимость кВт*ч, у.е.	Итого за год, у.е.
1	Компьютер комплектующими	1	220	240	8	0,33	633,6

Рассчитаем остальные затраты:

$$P_{rm} = 0,6 * K_{ob} = 380,16 \text{ у.е.}$$

$$P_{rem} = 1,3 * K_{ob} = 823,68 \text{ у.е.}$$

$$P_{zap} = 0,5 * K_{ob} = 316,80 \text{ у.е.}$$

$$P_{ks} = 70,60 * 12 = 847,20 \text{ у.е. (тарифы Мосэнергосбыт)}$$



$$P_{pr} = 0,6 * P_{obs} = 1800,86 \text{ у.е.}$$

В итоге получим:  $P_{obs} = 3490,00 \text{ у.е.}$ ,  $P = 9164,80 \text{ у.е.}$

Для того, чтобы определить экономическую эффективность ( $E_{ff}$ ), необходимо подсчитать расходы до и после разработки и внедрения РСОИ по следующей формуле (9):

$$E_{ff} = P_o - P_{lan} , \quad (9)$$

где  $P_o$  - расходы до разработки и внедрения РСОИ, у.е.;

$P_{lan}$  - расходы после разработки и внедрения РСОИ, у.е.

В процессе расчетов получим:  $P_o = 11032,80 \text{ у.е.}$

Следовательно:  $E_{ff} = P_o - P_{lan} = 1868,00 \text{ у.е.}$

Срок окупаемости разработанной системы ( $T_r$ ) рассчитывается как отношение капитальных затрат к экономической эффективности согласно формуле (10):

$$T_r = K/E_{ff}. \quad (10)$$

Получим:  $T_r = K/E_{ff} = 3,10$

Обратная величина будет представлять расчетный коэффициент приведения ( $E_r$ ) согласно формуле (11):

$$E_r = E_{ff}/K. \quad (11)$$

Получим:  $E_r = E_{ff}/K = 0,32$ .

Этот показатель необходимо сравнить с нормативным коэффициентом приведения ( $E_n = 0.25...0.35$ ). При соблюдении условия:  $E_r < E_n$  рассматриваемые инженерные решения признаются эффективными.

Поскольку условие  $E_r$  ( $E_r = 0,32 < E_n$  ( $E_n = 0.25...0.35$ )) соблюдается, то внедрение системы можно считать экономически эффективным.

## Заключение

Результатом данного проекта является создание программного средства для автоматизации торгового предприятия. Разработанная база данных предназначена для упрощения обработки информации, ввода и поиска данных о продукции. Создание базы данных необходимо для осуществления автоматизированного учета, хранения, извлечения и добавления информации о товаре, сотрудниках, поставщиках. К тому же, разработка собственного решения избавляет организацию от возникновения таких проблем, как прекращение технической поддержки или лицензирования готового продукта в связи с особенностями регионального расположения.

Достигнута цель выпускной квалификационной работы: создана автоматизированная система управления торговым предприятием (движение товаров на складе).

В процессе достижения цели были выполнены поставленные задачи:

- Проанализирована организационная структура.
- Проанализировано функционирование.
- Проанализировано использование информации.
- Спроектирована база данных.
- Разработан оконный графический интерфейс для базы данных.

Для проверки результатов эффективности внедрения системы автоматизации экономические показатели были рассчитаны по формулам. В связи с необходимостью сопоставить результаты внедрения ИС в виде экономии, и понесенные затраты – для анализа был взят показатель коэффициента цена/прибыль, чтобы использовать метод снижения затрат. По итоговой формуле коэффициентов  $E_r$  ( $E_r = 0,32$ )  $< E_n$  ( $E_n = 0.25...0.35$ ) эффективность была доказана.

В ходе работы над данным проектом были рассмотрены основные принципы разработки автоматизированной системы управления торговым предприятием. Был проведён анализ существующих информационных систем,

их основные функции и возможности, а также выявлены требования предприятия к системе автоматизации. Так же, было разработано техническое задание на создание информационной системы на основе анализа имеющихся решений и современных технологий.

В рамках данного проекта была автоматизирована значительная часть бизнес-процессов организации. Однако, после успешного внедрения системы, её дальнейшее развитие будет зависеть от потребностей предприятия. Предусмотрена не только возможность расширения функционала между отделами, но и интеграция с уже существующими реляционными базами данных, а также оптимизация графического интерфейса. Для корректного определения потребностей в будущем планируется сбор статистики и обратной связи, что поможет выстроить конкретный план действий, необходимых данному торговому предприятию.

## Список используемой литературы

- 1 Базы данных. В 2х кн. Кн. 2. Распределенные и удаленные базы данных: Учебник / В.П. Агальцов. М.: ИД ФОРУМ: НИЦ ИнфраМ, 2013. 272 с.: ил.; 60x90 1/16. (Высшее образование). (переплет) ISBN 9785819903940  
Режим доступа: <http://znanium.com/catalog/product/372740>.
- 2 Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. 2е изд., испр. и доп. М.: Форум: ИНФРАМ, 2007. 400 с.: ил.; 60x90 1/16. (Профессиональное образование). (переплет) ISBN 9785911340988 – Режим доступа: <http://znanium.com/catalog/product/126407>
- 3 Бейзер Б. Тестирование чёрного ящика. Технологии функционального тестирования программного обеспечения и систем. – СПб.: Питер, 2004. — 320 с.
- 4 Боггс, У. UML и Rational Rose / У. Боггс, М. Боггс. - М.: Издательство «ЛОРИ», 2001. - 582 с.
- 5 Бугорский, В.Н. Сетевая экономика и проектирование информационных систем : учеб.пос. / В.Н. Бугорский, Р.В. Соколов. СПб. : Питер, 2007. 320 с.
- 6 Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Дж. Рамбо, А. Якобсон. - СПб.: Питер, 2004. - 432 с.
- 7 Вагнер, Билл C# Эффективное программирование / Билл Вагнер. - М.: ЛОРИ, 2017. - 320 с.
- 8 Варжапетян, А.Г. Системы управления: Исследование и компьютерное проектирование : учеб.пос. / А.Г. Варжапетян, В.В. Глущенко. М. : Вузовская книга, 2005. 328 с.
- 9 Культин, Никита Основы программирования в Microsoft Visual C# 2010 / Никита Культин. - М.: БХВ-Петербург, 2017. - 389 с.
- 10 Ларман, К. Применение UML и шаблонов проектирования: Уч. Пос / К. Ларман. - М.: Издательский дом «Вильямс», 2001. - 496 с.

- 11 Леоненков, А.В. Самоучитель UML 2 / А.В. Леоненков. – СПб.: БХВ - Петербург, 2007. – 576с.
- 12 Малыхина, М. Базы данных: основы, проектирование, использование / М. Малыхина. - М.: БХВ-Петербург, 2016. - 512 с.
- 13 Редько, В.Н. Базы данных и информационные системы / В.Н. Редько, И.А. Басараб. - М.: Знание, 2017. - 341 с.
- 14 Ульман, Дж. Основы систем баз данных / Дж. Ульман. - М.: Финансы и статистика, 2017. - 292 с.
- 15 Фаулер, М. UML. Основы. Третье издание. / М. Фаулер. – М.: Символ-Плюс, 2006. – 192 с.
- 16 Microsoft SQL Server - URL: <https://www.techtarget.com/searchdatamanagement/definition/SQL-Server> (дата обращения 15.10.2023).
- 17 SQL Server – is a relational database management system (RDBMS) - URL: <https://www.sqlsplus.com/sql-server-is-a-relational-database-management-system-rdbms/> (дата обращения 15.10.2023).
- 18 What is a relational database? - URL: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-relational-database> (дата обращения 15.10.2023).
- 19 What Is SQL Server – Definition And Architecture - URL: <https://analyticsplanets.com/what-is-sql-server-definition-and-architecture-details/> (дата обращения 15.10.2023).
- 20 What is SQL Server- URL: <https://www.sqlservertutorial.net/getting-started/what-is-sql-server/> (дата обращения 15.10.2023).