

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)

Бизнес-информатика
(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка информационной системы учета товаров на складе»

Обучающийся

И.М. Маслов

(Инициалы Фамилия)

(личная подпись)

Руководитель

канд. экон. наук, доцент, Т.А. Раченко

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

Тольятти 2024

Аннотация

Тема бакалаврской работы – «Разработка информационной системы учета товаров на складе».

ИС складского учета является важным аспектом в эффективности управления любого предприятия, а применение соответствующего программного обеспечения, ориентированного под нужды именно этого предприятия с удобным и понятным интерфейсом позволяет оптимизировать работу с данными о товарах, ускоряя обработку документов, повышая точность учета продукции и другие аспекты управления складом.

Для достижения цели были выдвинуты следующие задачи: провести анализ предметной области с целью выявления возможностей для автоматизации существующих бизнес-процессов, проанализировать существующие информационные системы учета товаров на складе, сформулировать необходимые требования, разработать логическое и физическое проектирование системы, провести функциональное тестирование работы, а также оценить экономическую эффективность проекта.

В результате работы была разработана информационная система учета товаров на складе (на примере ООО «Энерготехмаш»), обеспечивающая удобное и эффективное управление операций учета товаров на складе, позволяя экономить время и ресурсы.

Объем работы содержит 76 страниц, на которых размещены 75 иллюстраций и 7 таблиц, 32 библиографических источника (5 из которых зарубежные).

Содержание

Введение.....	4
Глава 1 Анализ предметной области и постановка задач.....	6
1.1 Характеристика предметной области	6
1.2 Функциональное моделирование бизнес-процесса учета товаров на складе.....	9
1.3 Обзор и сравнительный анализ ИС для складского учета организаций	12
1.4 Определение требований и постановка задачи.....	16
1.5 Реинжиниринг бизнес-процесса учета товаров на складе.....	18
Глава 2 Логическое проектирование информационной системы учета товаров на складе.....	21
2.1 Разработка логической модели информационной системы	21
2.2 Информационное обеспечение ИС	25
2.3 Проектирование базы данных ИС	29
2.4 Требования к аппаратно-программному обеспечению ИС	37
Глава 3 Физическое проектирование ИС.....	40
3.1 Критерии выбора архитектуры и технологии ИС	40
3.2 Разработка ИС учета товаров на складе	43
3.3 Функциональное тестирование ИС	53
3.4 Расчет показателей экономической эффективности проекта.....	66
Заключение	72
Список используемой литературы	74

Введение

В современном обществе информационные технологии стремительно развиваются, и одним из секторов, где они широко применяются, является область складского хозяйства.

Системы складского учета обеспечивают надежное хранение всей информации о товарах, начиная с их поступления на склад и заканчивая окончательной отгрузкой. Эти системы не только упрощают процесс учета, но и существенно снижают вероятность человеческих ошибок, увеличивают прозрачность операций и повышают общую эффективность бизнеса, что имеет критическое значение.

Актуальность данной темы заключается в том, что современные компании сталкиваются с необходимостью более оперативного и точного управления запасами, что диктует необходимость внедрения продвинутых информационных систем для оптимизации складских операций.

Цель данной работы заключается в разработке информационной системы учета товаров на складе.

Объектом исследования бакалаврской работы является складская логистика и процессы учета готовой продукции на складе в организации ООО «Энерготехмаш».

Предметом исследования бакалаврской работы является информационная система, предназначенная для автоматизации процессов учета готовой продукции на складе.

Задачами исследования являются:

- характеристика предметной области;
- функциональное моделирование бизнес-процесса учета товаров на складе;
- реинжиниринг бизнес-процесса учета товаров на складе;
- логическое проектирование информационной системы учета товаров на складе;

- физическое проектирование ИС;
- функциональное тестирование ИС;
- расчет показателей экономической эффективности проекта.

Бакалаврская работа состоит из трех глав, заключения и списка используемой литературы.

Первая глава служит основой для понимания предметной области, с подробным описанием характеристик организации и исследуемого бизнес-процесса. Также в аналитической части рассмотрены существующие программные продукты, предназначенные для ведения складского учета.

Во второй главе рассматривается проектирование системы с выполнением концептуального моделирования и разработок требований к разрабатываемой системе, разработки логической модели информационной системы, построением диаграммы вариантов использования системы и диаграммы классов, преобразованной в ER-диаграмму базы данных.

В третьей главе проведено физическое проектирование информационной системы с акцентом на выборе архитектуры, системы управления базами данных (СУБД) и детальном изучении существующих технологий разработки программного обеспечения, чтобы выбрать наиболее подходящую для наших целей. В завершении проведено тестирование программного проекта и расчеты показателей экономической эффективности.

Глава 1 Анализ предметной области и постановка задач

1.1 Характеристика предметной области

Предприятие ООО «Энерготехмаш» занимает лидирующие позиции среди производителей и поставщиков алюминиевого профиля в России.

Основное направление деятельности – производство экструдированных профилей, штампованных металлических изделий и алюминиевых профильных систем. Площадь производственных помещений превышает 90000 м² и включает несколько основных направлений: алюминиевое, прессовое, матричное, производство автокомпонентов и полимерное производство. Основная концепция предприятия заключается в индивидуальном подходе к каждому клиенту, уделяя внимание всем деталям и требованиям, представленных в чертежах, с целью производства продукции, отвечающей ожиданиям заказчика, контролируя качество на всех этапах производства. На ООО «Энерготехмаш» организационная структура является линейно-функциональной, она представлена на рисунке 1.

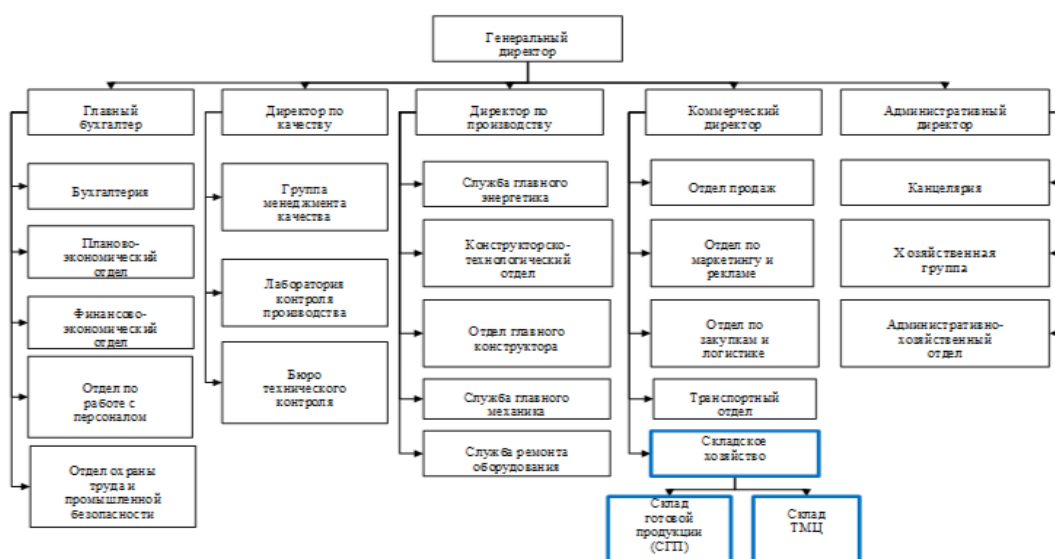


Рисунок 1 – Организационная диаграмма ООО «Энерготехмаш»

Все подразделения в организации не могут функционировать независимо друг от друга. Для более наглядного представления этих связей и взаимодействий создана блок-схема, изображенная на рисунке 2.

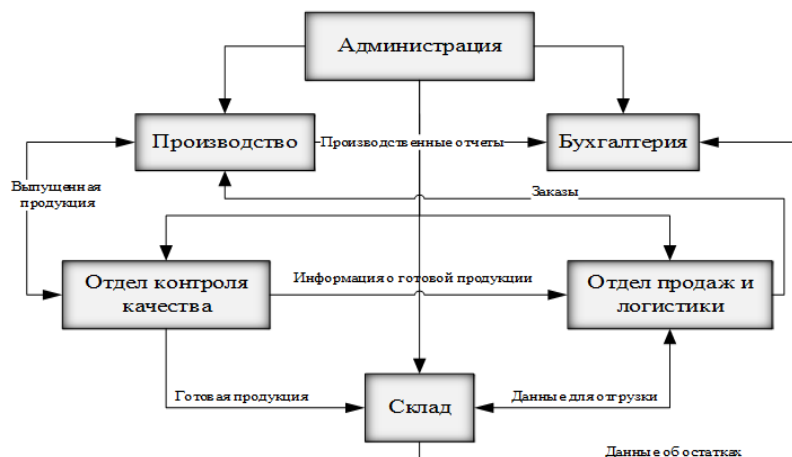


Рисунок 2 – Блок-схема взаимодействия между отделами предприятия ООО «Энерготехмаш»

Блок-схема наглядно показывает взаимосвязи между подразделениями предприятия, отображая их взаимодействие между собой:

- администрация играет ключевую роль в координации всех операций, обеспечивая стратегическое руководство и поддержку всем остальным отделам.
- Бухгалтерия, в свою очередь, занимается финансовым учетом и контролем, представляя данные о расходах и доходах, что необходимо для принятия управленческих решений.
- Производственный отдел отвечает за создание готовой продукции и находится в постоянном взаимодействии с отделом контроля качества, который следит за соблюдением стандартов.
- Отдел продаж, принимая участие в формировании спроса, активно работает с логистикой и складом для оптимизации процессов доставки и хранения товаров [10].

- Логистика управляет движение продукции, гарантируя своевременные поставки и минимизацию издержек.
- Склад является важным звеном в логической цепи, интегрируя взаимодействие между производственными и сбытовыми подразделениями компании.

Для выполнения задачи было выбрано подразделение складского хозяйства склад готовой продукции (далее СГП).

Отношения и взаимодействие склада готовой продукции (СГП) с подразделениями осуществляется согласно организационной структуре управления предприятием, выполняемыми задачами и функциями [9].

Структура подразделения представлена на рисунке 3.



Рисунок 3 – Структура подразделения склада ООО «Энерготехмаш»

Контроль за деятельностью склада в компании возложен на начальника складского хозяйства, который играет ключевую роль в обеспечении эффективной логистики и бесперебойной работы. Основные обязанности начальника складского хозяйства включают в себя следующее:

- организация и координирование процессов приема и хранения товаров, с разработкой оптимальных схем размещения для эффективного использования площадей. Это способствует экономии места и ускоряет процессы подбора и комплектации заказов.

- Контроль своевременной подготовки товаров к отгрузке клиентам организации.
- Проверка отгружаемых товаров на наличие необходимой маркировки, что обеспечивает соответствие стандартам и требованиям клиентов.
- Обеспечение сохранности потребительских свойств товаров.

Процесс работы склада схематично изображен на рисунке 4.

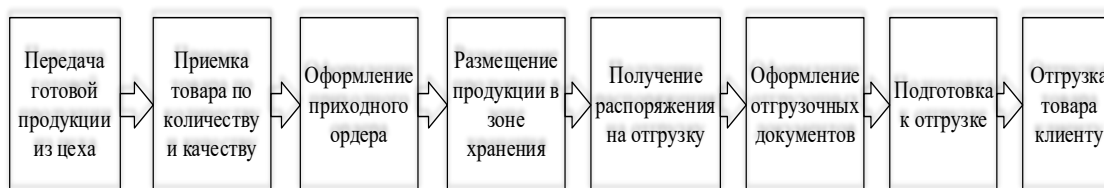


Рисунок 4 – Процесс работы склада ООО «Энерготехмаш»

Хранение и учет товаров в компании являются ключевыми процессами и поэтому отдел бухгалтерии играет важную роль в контроле за движением товаров на складе. Бухгалтерия осуществляет контроль за операциями на складе, проводит проверку документов по учету товаров и прочее. Поэтому внедрение информационной системы поможет упростить работу как сотрудникам склада, так и бухгалтерскому отделу.

1.2 Функциональное моделирование бизнес-процесса учета товаров на складе

Разработка информационной системы начинается с построения модели работы организации, существующий на данный момент («Как есть»). Анализ модели позволяет понять, где находятся слабые места, в чем будут состоять преимущества новых процессов и насколько глубоким изменениям подвергнется существующая организация деятельности предприятия [2].

На рисунке 5 показана контекстная диаграмма концептуальной модели работы склада готовой продукции ООО «Энерготехмаш» («Как есть») в нотации IDEF0.

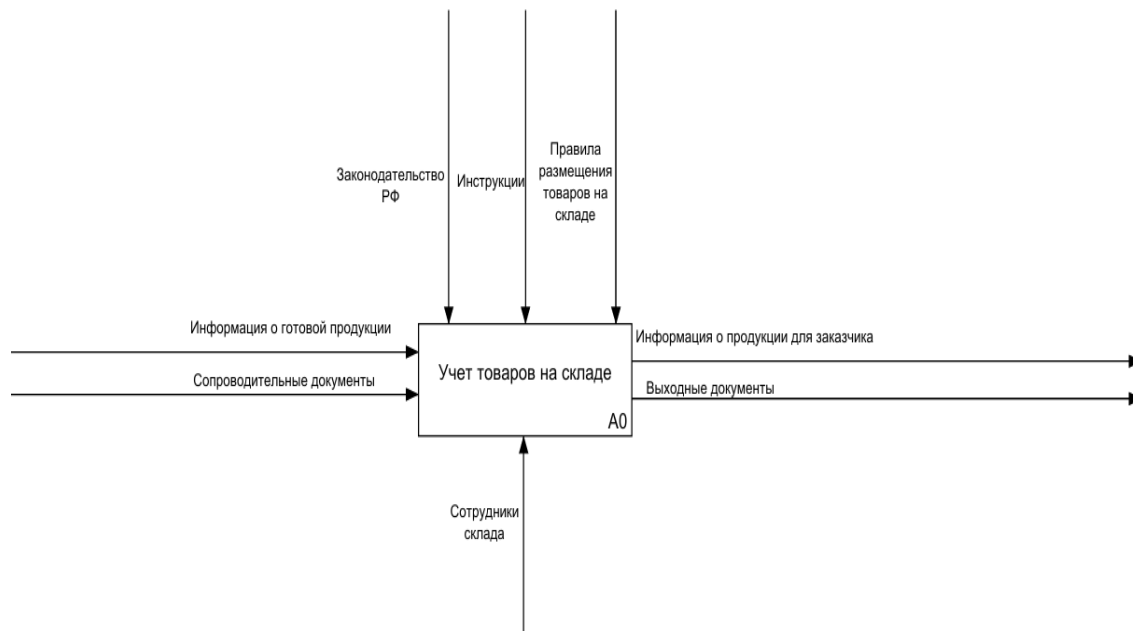


Рисунок 5 – Контекстная диаграмма процесса учета товаров на складе «Как есть»

Прием продукции на склад, сопровождается соответствующими приходными документами. В них указывается дата, код цеха и склада, название продукции, единица измерения, количество, номенклатурный номер.

Поступившая для хранения на складе продукция принимается на ответственное хранение, для чего выделяются подходящие зоны хранения (склад делится на несколько зон, каждая из которых может вместить некоторое количество продукции). В ходе этого процесса кладовщиком создается документ «Приходный ордер», в котором поступившая продукция приходуется и распределяется по зонам склада. В существующем процессе кладовщику приходится оценивать занятость зон на месте, так как оценка по бумагам занимает слишком много времени.

При поступлении заказа, менеджером формируется документ «Распоряжение на отгрузку, с учетом наличия заказной продукции на складе. При получении документа «Распоряжения на отгрузку», кладовщиком оформляется документ «Расходный ордер», товар укомплектовывается в соответствии с требованиями заявки и отгружается.

На рисунке 6 представлена декомпозиция первого уровня процесса учета товара на складе.

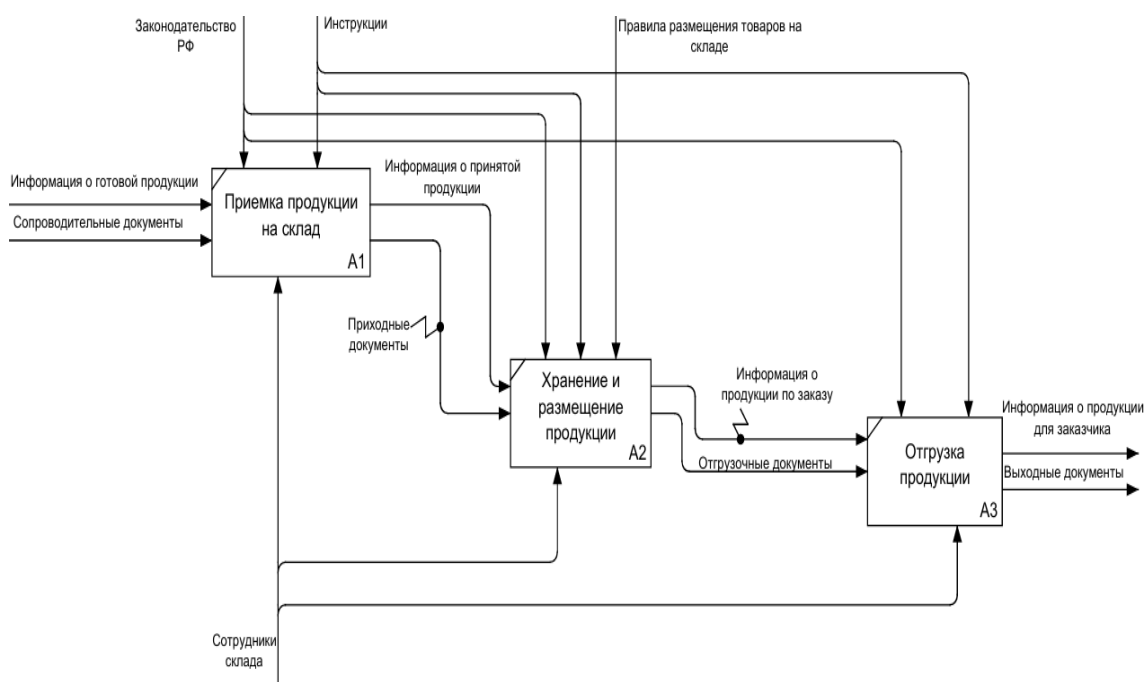


Рисунок 6 – Декомпозиция первого уровня процесса учета товаров на складе «Как есть»

Анализ существующей модели бизнес-процесса позволяет выявить «узкие места» модели, которые можно устранить с помощью внедрения автоматизированной информационной системы. Учет товаров на основе бумажных документов является крайне неэффективным, приводит к ошибкам и неточностям при составлении документов, извлечение какой-либо структурированной информации на основе бумажных документов (например, ведомость по товарам на складе) является очень трудоемким процессом [23].

1.3 Обзор и сравнительный анализ ИС для складского учета организаций

На сегодняшний день существует множество готовых информационных систем для решения задач учета товаров на складе. Готовые информационные системы часто предлагают уже отработанные функционалы, которые не всегда можно адаптировать под специфические потребности компании [13].

Проведем анализ производительности, надежности и поддержки у существующих решений, опираясь на оценки отзывов пользователей и кейсов внедрения. В качестве критериев изучения готовых решений учтем следующие параметры:

- наличие пробной версии;
- функциональность;
- удобство интерфейса;
- гибкость и простота настройки;
- соответствие с функционалом компании;
- возможность интеграции с другими системами;
- наличие технической поддержки и обновлений;
- экономическая целесообразность [20].

Большой популярностью пользуются системы «1С: Логистика: Управление складом 3.0», «Мой склад» от компании «Логнекс» и «1С: Комплексная автоматизация». Данные решения представляют собой эффективные инструменты для управления логистическими процессами и оптимизации учета товаров на складе.

«1С: Логистика: Управление складом 3.0» подходит для разных типов складов, включая небольшие и крупные распределительные центры. Система позволяет учитывать несколько складов и зон хранения, определять местонахождение и количество товаров, с детальной информацией о сроках годности, партиям и серийным номерам.

Основные возможности ИС «1С: Логистика: Управление складом 3.0» представлены на рисунке 7.



Рисунок 7 – Возможности ИС «1С: Логистика: Управление складом 3.0»

Начальная страница системы «1С: Логистика: Управление складом 3.0» представлена на рисунке 8.

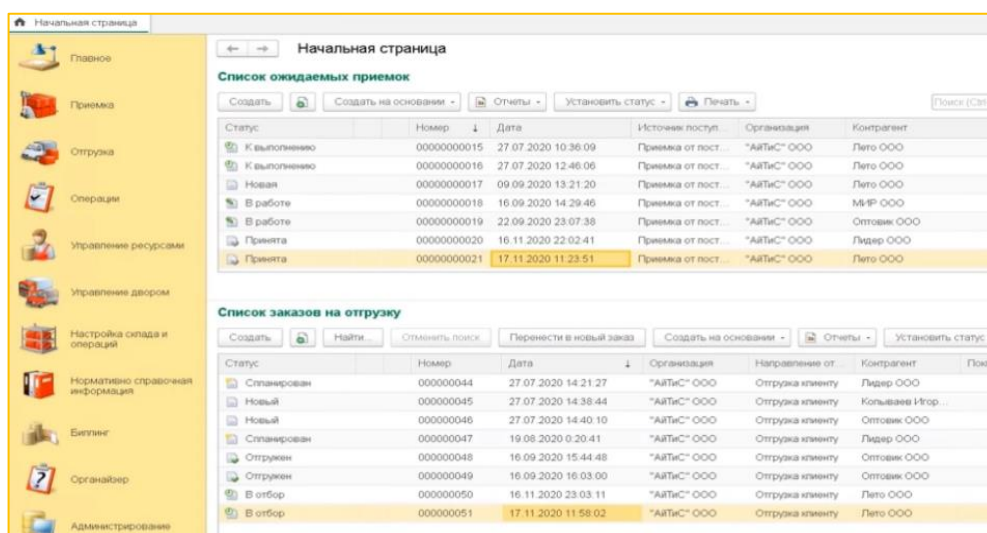


Рисунок 8 – Начальная страница ИС «1С: Логистика: Управление складом 3.0»

Облачное приложение «Мой склад» представляет собой полнофункциональный бэк-офис, способный обслуживать малые и средние предприятия. Оно помогает легко вести учет продукции, анализировать прибыльность товаров и отслеживать остатки на складе.

В данной программе возможно осуществлять различные операции со складом: приемку и отгрузку товаров, проведение инвентаризации и учет остатков (рисунок 9).

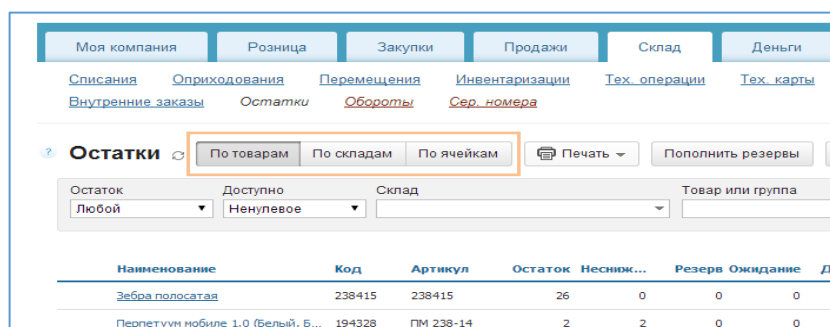


Рисунок 9 – Модуль складской учет

Среди функциональных возможностей системы: складской учет, интеграция с различным оборудованием, адресное хранение и возможность печати документов.

«1С: Комплексная автоматизация» – это программа, которая автоматизирует ключевые процессы компании: бухгалтерию, складской учет, управление кадрами и так далее.

Этот софт подходит небольшим торговым компаниям, которые хотят контролировать все процессы в одной программе, а также производственным предприятиям с собственным складом, где не нужно регистрировать заказы.

Основной функционал программы:

- приемка и перемещение;
- инвентаризация;
- сбор заказов;

– отгрузка.

Подраздел программы для учета товаров на складе представлен на рисунке 10.

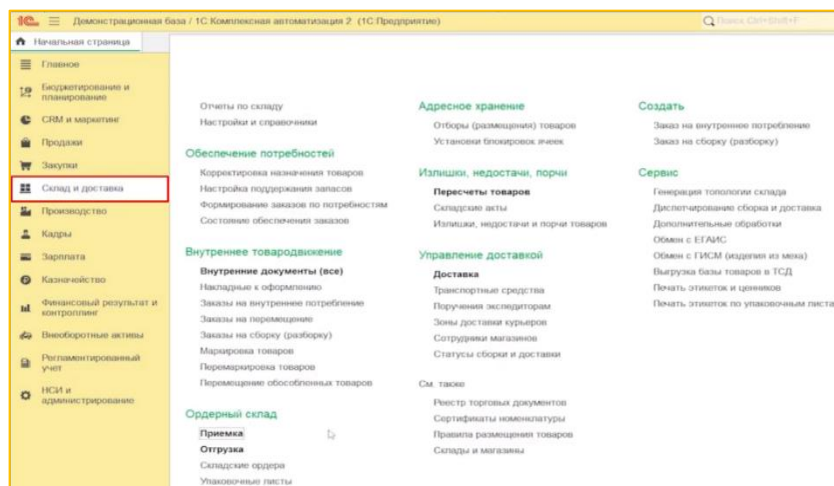


Рисунок 10 – Подраздел «Склад и доставка»

В таблице 1 представлен сравнительный анализ функциональных возможностей описанных программных продуктов ИС.

Таблица 1 – Сравнительный анализ функциональных возможностей

Функциональные возможности	1С: Логистика: Управление складом 3.0	Мой склад	1С: Комплексная автоматизация
Наличие пробной версии	+	+	-
Функциональность	+	+	+
Удобство интерфейса	-	+	+
Соответствие с функционалом компании	+	+	+
Гибкость и простота настройки	+	-	-/+
Возможность интеграции с другими системами	+	+	+
Наличие технической поддержки и обновлений	+	+	+
Экономическая целесообразность	-	-	-

При сравнительном анализе были обнаружены преимущества и недостатки каждой системы. Каждое из этих решений имеет свои уникальные особенности и подходят для разного вида бизнеса.

«1С: Логистика: Управление складом 3.0» обладает широким спектром функций, что делает его идеальным выбором для средних и крупных предприятий. Интерфейс программы имеет сложную и непонятную структуру.

«Мой склад» подойдет малым компаниям благодаря простоте использования и конкурентоспособной цене, его возможности интеграции и настройке несколько ограничены.

«1С: Комплексная автоматизация» представляет собой широкие функциональные возможности, но может потребовать значительных усилий для настройки и освоения. Высокая стоимость оправдана функционалом, которой охватывает все аспекты бизнеса.

Исходя из анализа характеристик готовых программ и сравнения их между собой, было решено разработать индивидуальное решение информационной системы для учета готовой продукции на складе.

1.4 Определение требований и постановка задачи

Для разработки требований мы применим методологию FURPS+, для определения основных требований, касающихся функциональности каждого модуля учета товаров на складе.

Аббревиатура FURPS означает «функциональность, удобство использования, надежность, производительность и возможность поддержки» (англ. Functionality, Usability, Reliability, Performance, Supportability). В методологии FURPS+ требования к ПО разделены на функциональные и нефункциональные требования [12].

Таблица 2, построенная с учетом методологии FURPS+, содержит структурированный набор требований для каждого аспекта.

Таблица 2 – Ключевой набор требований к программному обеспечению

Категории	Подкатегории	Описание
Функциональные требования (Functionality)	Хранение данных о товарах, заказчиках и зонах хранения	Функция управления пользователями системы
		Функция управления номенклатурой товаров
		Функция управления заказчиками
		Функция управления сотрудниками
	Хранение и обновление информации о приходе готовой продукции на склад	Регистрация поступления товара
	Хранение и обновление информации о размещении товаров на складе	Управление процессом размещения товаров на складе
		Отображение текущего статуса и местонахождение каждого товара
	Отслеживание отгрузки товаров и остатков	Рационализация процессов отгрузки товаров с возможностью отслеживания статусов отгрузки
	Управление базой данных (добавление, редактирование, удаление)	Разработать интерфейс и механизмы для безопасного добавления, редактирования и удаления записей.
	Создание документов о складских операциях	Автоматизация процесса формирования документов для складских операций (накладные, ведомости и так далее)
	Генерация и печать отчетов	Печать отчетов и документов непосредственно из интерфейса системы
Требования к удобству использования (Usability)	Удобный графический интерфейс пользователя	Приятный для глаз и интуитивно понятный интерфейс
		Легкость навигации и выполнения основных операций
	Настройка профилей пользователей с различными уровнями доступа	Роли и права доступа в зависимости от обязанностей (например, кладовщик, менеджер, администратор)
	Отсутствие функциональной избыточности	Только необходимые функции, чтобы не перегружать пользователей ненужными опциями
Требования к надежности (Reliability)	Вход разрешен только зарегистрированным пользователям	С возможностью определения ролей и доступа к определенным действиям для каждого пользователя

Продолжение таблицы 2

Требования к производительности (Performance)	Минимизация времени отклика на запросы пользователей	Обеспечение оперативной обработки и обновления данных при проведении операций по учету товаров на складе
Требования к поддержке (Supportability)	Легкое расширение функционала системы	Модульная архитектура, позволяющая добавлять новые функции без необходимости полной перестройки системы

Эта таблица представляет четкую структуру требований и поможет эффективно планировать и реализовывать проект.

1.5 Реинжиниринг бизнес-процесса учета товаров на складе

В ходе анализа бизнес-процессов ООО «Энерготехмаш» была разработана оптимизированная модель «Как должно быть» в нотации IDEF0. Это сделано для автоматизации и улучшения процессов приема, хранения и отгрузки продукции [16].

Новая модель основана на текущей модели «Как есть» и представлена на контекстной диаграмме на рисунке 11.

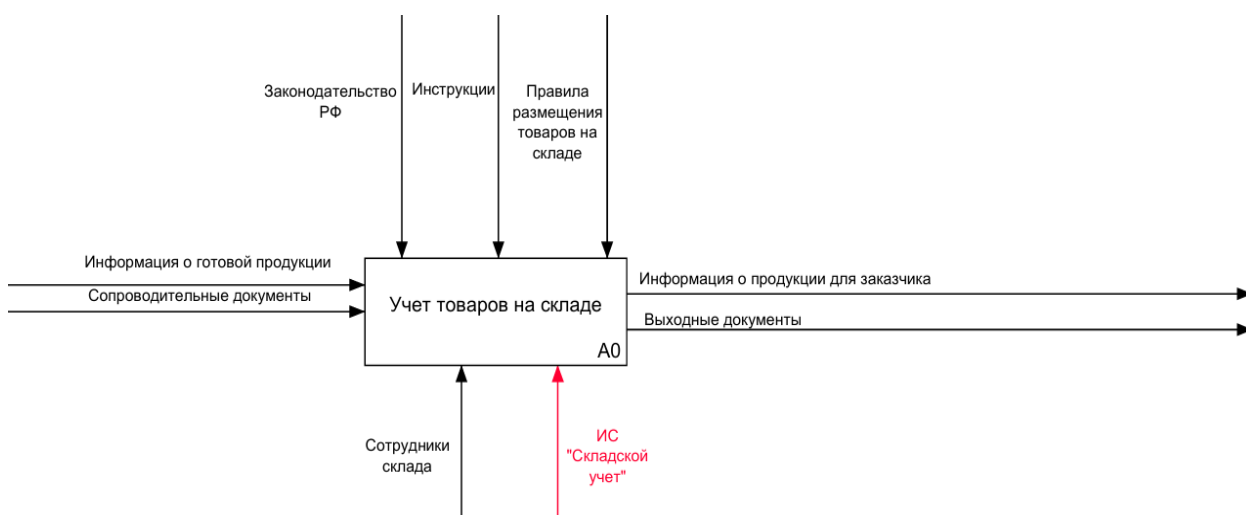


Рисунок 11 – Контекстная диаграмма бизнес-процесса учета товаров на складе «Как должно быть»

Произведем декомпозицию контекстной диаграммы (рисунок 12).

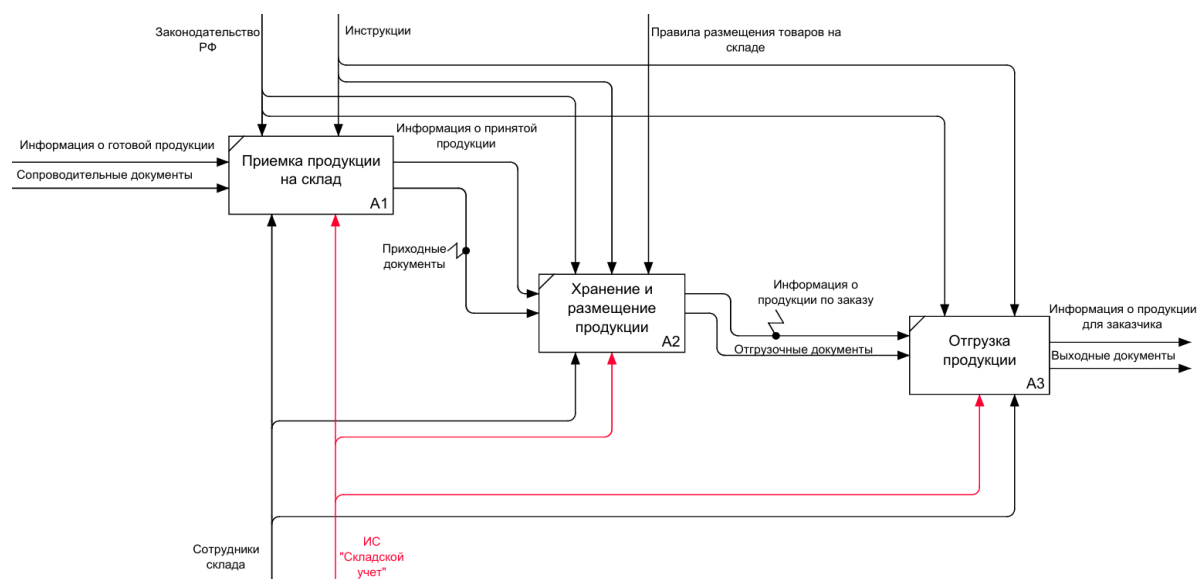


Рисунок 12 – Диаграмма декомпозиции бизнес-процесса учета товаров на складе «Как должно быть»

Эта диаграмма содержит три подпроцесса, подробности которых приведены в таблице 3.

Таблица 3 – Подпроцессы диаграммы декомпозиции

Подпроцесс	Вход	Процесс	Выход
Приемка продукции на склад	Информация о поступившей готовой продукции, сопроводительные документы	Ввод информации о поступлении в систему	Обновленные записи в системе учета, отчет о приемке
		Создание записи о поступлении в базе данных	
Хранение и размещение продукции	Информация о принятой продукции	Определение места хранения продукции	Обновленные записи в системе учета
		Обновление записи в системе учета о местонахождении продукции	
Отгрузка продукции	Информация о продукции по заказу, отгрузочные документы	Ввод информации об отгрузке в систему	Обновленные записи в системе учета, отгрузочные документы
		Генерация отгрузочных документов	

После завершения анализа функциональных требований к предметной области, можно приступать к проектированию информационной системы учета товаров на складе для ООО «Энерготехмаш».

Выводы по главе 1.

В первой главе проведен анализ предметной области, рассмотрена организационная структура и описаны процессы учета готовой продукции на складе.

Проведено моделирование бизнес-процессов «Как есть» и «Как должно быть» в нотации IDEF0, что позволило понять, где находятся слабые места, которые можно устранить с помощью внедрения автоматизированной информационной системы. Определение требований и рассмотрение предлагаемых программных решений для складского учета привели к необходимости разработки собственной ИС для учета товаров на складе. Это решение объясняется тем, что софт предлагаемых программных решений избыточен и сложен в использовании для небольшого склада.

Глава 2 Логическое проектирование информационной системы учета товаров на складе

2.1 Разработка логической модели информационной системы

Логическое моделирование может быть выполнено с использованием различных методологий, наиболее популярные из которых основаны на нотациях IDEF и UML. В данной работе для логического проектирования ИС используется нотация UML [31].

UML (Unified Modeling Language) – унифицированный язык моделирования – это семейство графических нотаций, в основе которого лежит единая метамодель. Он помогает в описании и проектировании программных систем, в особенности систем, построенных с использованием объектно-ориентированных технологий [8].

К преимуществам языка UML можно отнести разнообразные инструментальные средства, которые, как поддерживают жизненный цикл ИС, так и позволяет настроить и отразить специфику деятельности разработчиков различных элементов проекта [5].

Диаграмма вариантов использования представляет собой графическое отображение взаимодействия пользователей с системой, позволяя визуализировать функции и требования к программному обеспечению.

В центре внимания находятся actores (участники), которые могут быть, как внешними пользователями, так и другими системами, взаимодействующими с проектным решением [1].

В соответствии с поставленной задачей, на диаграмме изображены три типа пользователей – «Администратор», «Кладовщик», «Менеджер» и основные операции, доступные для каждой роли пользователя.

На рисунке 13 продемонстрирована диаграмма вариантов использования, как одна из основных диаграмм, относящихся к UML.



Рисунок 13 – Диаграмма вариантов использования ИС

Отметим, что только «Администратор» имеет прямой доступ ко всем таблицам и базе данных [29].

Ключевые моменты для роли «Администратор», входящие в администрирование ИС:

- добавление/удаление/изменение пользователей системы;
- редактирование таблиц базы данных;
- устранение неисправностей.

На стадии анализа требований и начального проектирования системы на основе построенных моделей системных прецедентов создаются диаграммы классов системы. На диаграмме классов могут быть указаны внутренняя структура и типы отношений между отдельными объектами и подсистемами, что приводит к развитию концептуальной модели системы [6].

Диаграмма классов на рисунке 14 представляет структуру информационной системы для складского учета товаров в ООО «Энерготехмаш».

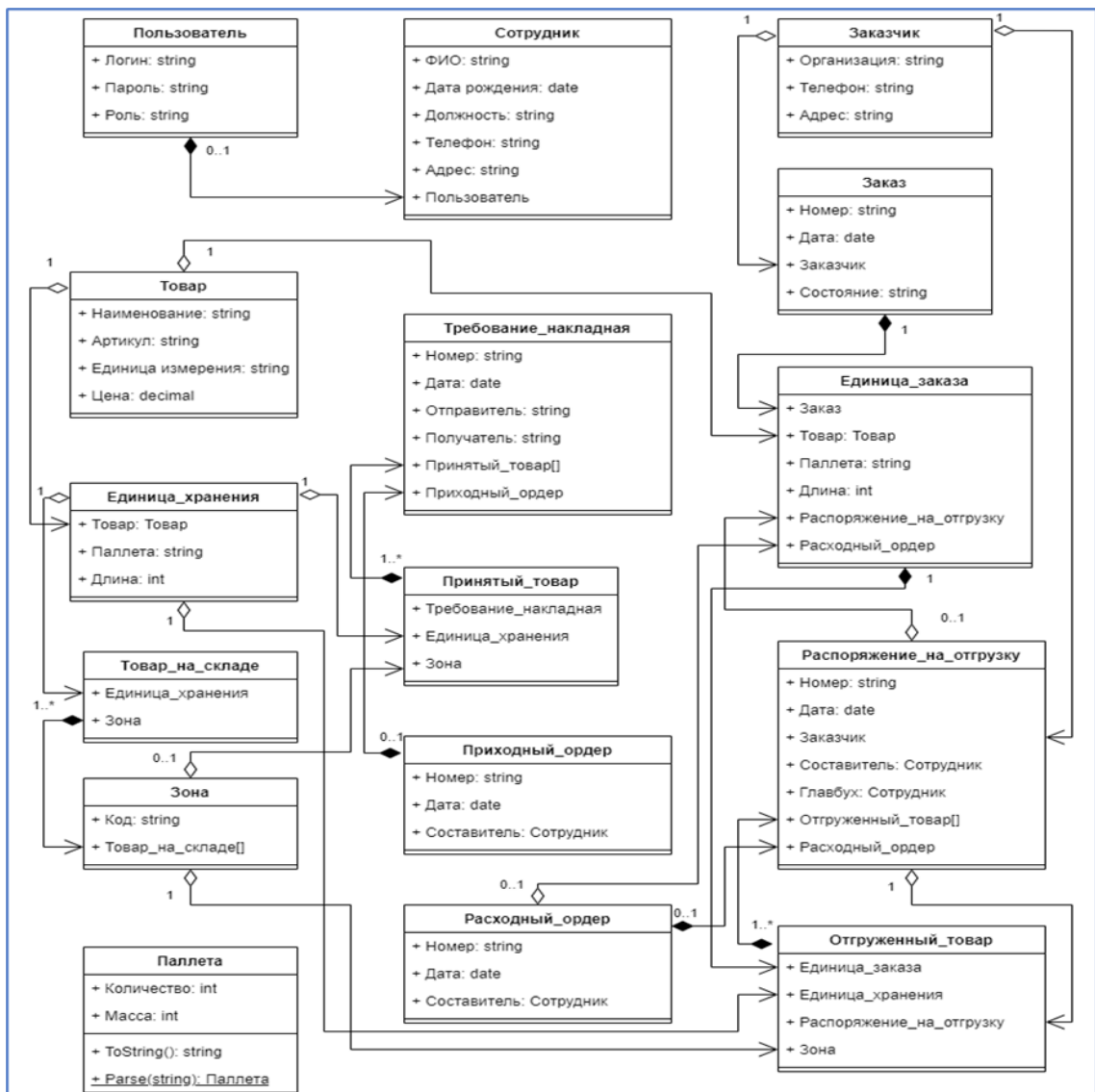


Рисунок 14 – Диаграмма классов информационной системы

Класс «Пользователь» предназначен для хранения всех зарегистрированных в системе пользователей.

Класс «Сотрудник» соответственно предназначен для хранения информации обо всех сотрудниках склада. Оба класса связаны отношением композиции – один пользователь может быть связан только с одним сотрудником (на это указывает закрашенный ромб). Сотрудник может быть пользователем системы, но может и не быть (на это указывает кратность «0..1» на диаграмме).

Классы «Заказчик» и «Заказ» связаны отношением агрегации – заказчик может быть связан со многими заказами (не закрашенный ромб). Каждый заказ обязательно связан с определенным заказчиком (кратность «1»).

Класс «Товар» предназначен для хранения информации о номенклатуре товаров.

Класс «Единица хранения» имеет ссылку на конкретный товар и описывает параметры базовой единицы хранения – паллета (шт./кг) и длина.

Похожий класс «Единица заказа» помимо этого имеет ссылки на заказ, распоряжения на отгрузку и расходный ордер. При приеме заказа единица заказа сразу связывается с заказом (поэтому кратность для их связи «1»). Две другие ссылки изначально пустые, так как соответствующие документы еще не созданы. Назначение этих ссылок служить маркерами состояния единицы заказа – изначально единица находится в состоянии «Ожидает распоряжения на отгрузку». После того, как единица попадает в распоряжение на отгрузку и соответствующее поле станет указывать на объект, состояние единицы заказа будет рассматриваться как «Ожидает отгрузки». После того, как данная единица попадает в расходный ордер и соответствующее поле станет указывать на объект, состояние единицы заказа будет рассматриваться как «Отгружен».

Класс «Требование_накладная» хранит информацию о товарах, принятых на склад по требованию-накладной, для чего в классе имеется список на объекты класса «Принятый_товар», который связывает базовую единицу хранения с требованием-накладной, а также с зоной, куда был определен товар. Зона изначально пуста и заполняется после создания объекта «Приходный_ордер», который также связывается с объектом «Требования_накладная».

Назначение таких сложных взаимоотношений между классами в том, чтобы полностью сохранить в системе все выполняемые операции.

Аналогично работает связка из трех классов «Распоряжение_на_отгрузку», «Отгруженный товар» и «Расходный_ордер», но еще присутствует связь с единицами заказа.

Классы «Зона» и «Товар_на_складе» предназначены для хранения информации о товарах, находящихся на складе в данный момент.

Класс «Паллета» является вспомогательным и предназначен для преобразования текстового представления паллеты в виде «XXXшт.YYY» в числовое представление и обратно.

Создание диаграмм в данной работе выполнялось с помощью инструментальных средств ConceptDraw DIAGRAM, Draw.io и Lucidchart.

2.2 Информационное обеспечение ИС

Основным входным документом системы является требование-накладная (форма № МХ-18 «Накладная на передачу готовой продукции»). Данные из этого документа вводятся кладовщиком при приеме товара в соответствующей экранной форме. В базе данных сохраняются такие атрибуты документа, как номер и дата, организация, отправитель, получатель, кто затребовал и кто получил, а также список товаров. Другим входным документом является «Заказ». Данные из этого документа вводятся менеджером при приеме заказа. В базе данных сохраняются такие атрибуты документа, как номер и дата, организация, а также список товаров.

Также есть три типа внутренних документов:

- «Приходный ордер» составляется кладовщиком на основе требования-накладной, с указанием зоны хранения, куда должна быть помещена каждая единица полученного товара. В базе данных сохраняется номер и дата составления документа, ФИО сотрудника, который оформил документ, а также ссылка на требование-накладную, на основе которой был создан этот ордер (рисунок 15).

Приходный ордер № _____ от _____

Отправитель: _____

Склад: _____

Распоряжение: _____

Входящий документ: _____

№ п/п	Зона	Артикул	Товар	Паллета	Ед. изм.	Длина, мм	Кол-во, шт

Всего _____ наименований

Отпустил _____ (подпись) _____ (должность) _____ (Ф.И.О.)

Получил _____ (подпись) _____ (должность) _____ (Ф.И.О.)

Рисунок 15 – Формат документа «Приходный ордер»

- «Распоряжение на отгрузку» составляется менеджером в процессе обработки заказов для передачи кладовщику и начала процесса комплектации товаров на отгрузку. В базе данных сохраняется номер и дата составления документа, информация о заказчике, ФИО сотрудника, который разрешил отгрузку, ФИО главного бухгалтера, а также список заказных товаров (рисунок 16).

Расходный ордер № _____ от _____

Распоряжение: _____

Склад: _____

Поставщик: _____

Покупатель: _____

№ п/п	Зона	Артикул	Товар	Заказ	Паллета	Ед. изм.	Длина, мм	Кол-во, шт

Всего _____ наименований

Отпустил _____ (подпись) _____ (должность) _____ (Ф.И.О.)

Получил _____ (подпись) _____ (должность) _____ (Ф.И.О.)

Рисунок 16 – Формат документа «Расходный ордер»

- «Расходный ордер» составляется кладовщиком на основе распоряжения на отгрузку, с указанием зоны, из которой должна быть взята каждая единица товара. В базе данных сохраняется номер и дата составления документа, ФИО сотрудника, который оформил документ, ФИО главного бухгалтера, а также ссылка на распоряжение на отгрузку, на основе которого был создан этот ордер (рисунок 17).

Распоряжение на отгрузку № _____ от _____						
Заказчик: _____						
Разрешил: _____						
Начальнику отдела отгрузки отгрузить: Спецификация на отгрузку со склада готовой продукции						
№ п/п	Товар	Заказ	Паллета	Ед. изм.	Длина, мм	Кол-во, шт
Всего по спецификации _____ наименований _____ мест						
Главный бухгалтер		_____	_____			
		(подпись)	(Ф.И.О.)			
Кладовщик		_____	_____			
		(подпись)	(Ф.И.О.)			
Исполнитель		_____	_____			
		(подпись)	(Ф.И.О.)			

Рисунок 17 – Формат документа «Распоряжение на отгрузку»

В процессе работы система может генерировать отчеты двух видов:

- «Ведомость по товарам на складе», которая отображает приход и расход для всех товаров сразу за определенный промежуток времени (рисунок 18).

Ведомость по товарам на складе

Параметры:

Период:

Склад:

Параметры: Период:

Склад:

Артикул	Товар	Ед. изм.	Начальный остаток	Приход	Расход	Конечный остаток

Рисунок 18 – Формат отчета «Ведомость по товарам на складе»

Типовая межотраслевая форма № М-17
Утверждена постановлением Госкомстата России от 30.10.97 № 71а

КАРТОЧКА № _____ учета материалов

Организация _____
Структурное подразделение _____

Форма по ОКСТД
_____ по ОКПО
Дата составления _____

Коды	
0315008	
65478124	

Структурное подразделение	Вид деятельности	Склад	Место хранения		Марка	Сорт	Профиль	Размер	Номенклатурный номер	Единица измерения		Цена, руб. коп.	Норма запаса	Срок годности	Поставщик
			стеллаж	ячейка						код	наименование				

Наименование материала без покрытия

Драгоценный материал (металл, камень)								номер паспорта	
наименование	вид	номенклатурный номер	единица измерения		количество (масса)	номер документа	наименование		
			код	наименование					
1	2	3	4	5	6	7	8	9	

Дата записи	Номер		От кого получено или кому отпущено	Учетная единица выпуска продукции (работ, услуг)	Приход	Расход	Остаток	Подпись, дата
	документа	по порядку						
1	2	3	4	5	6	7	8	9

Карточку заполнил _____

(должность) _____

(подпись) _____

(расшифровка подписи) _____

" " "

20 г.

Организация	_____	по ОКПО	65478124
	_____	Дата составления	_____

Структурное подразделение _____

[illegible]

Наименование материала без покрытия

[illegible][illegible]

Карточку заполнил _____ (должность) _____ (подпись) _____ (расшифровка подписи)

20

Рисунок 19 – Формат отчета «Карточка учета товара»

2.3 Проектирование базы данных ИС

При проектировании базы данных информационной системы модель классов преобразуется в модель реляционную, позволяющую связывать сущности и задавать рекурсивные связи [15].

Для описания модели базы данных принято использовать ER-модель (Entity-Relationship model, модель «сущность-связь»), которая может быть визуализирована с использованием различных нотаций [30].

На рисунке 20 изображена ER-диаграмма базы данных, изображенная в нотации «crow's foot» («воронья лапка»).

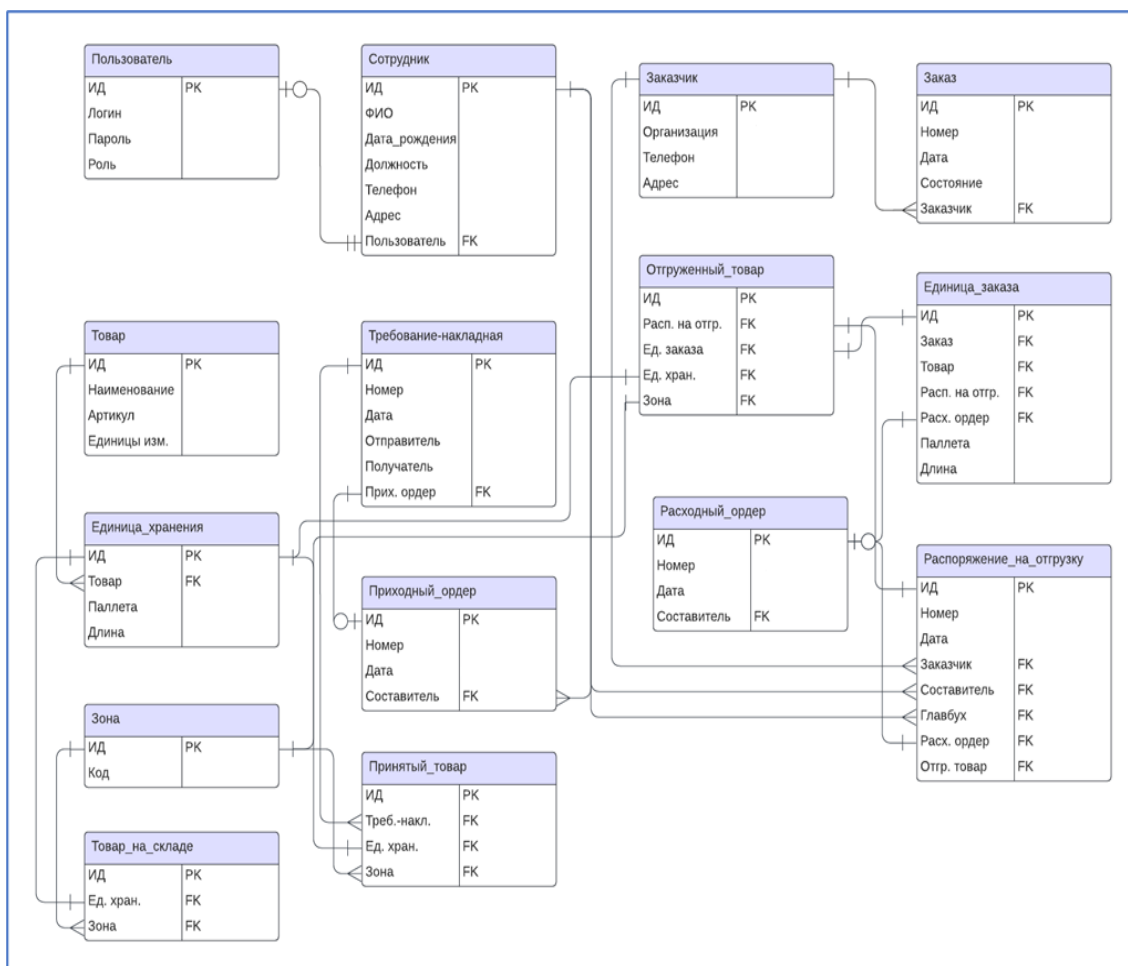


Рисунок 20 – Логическая модель базы данных в нотации «crow's foot» («воронья лапка»)

Сущности в данной нотации представлены в виде прямоугольников, содержащих строки – атрибуты сущности. Для каждого атрибута создается имя и тип хранимого значения [32].

В данной диаграмме используется несколько типов значений: РК (Primary Key) – первичный ключ, FK (Foreign Key) – внешний ключ [28].

Связи на диаграммах данной нотации помечаются специальными знаками (из-за чего данная нотация и получила свое название).

Вертикальная черта означает «один», кружок «ноль», «воронья лапка» означает «много».

Например, связь «Сотрудник-Пользователь» на стороне сотрудника помечена двумя чертами, что означает «один и только один», то есть определённый пользователь может быть связан только с одним сотрудником.

На стороне пользователя связь помечена чертой и кружком, что означает, что сотрудник может быть связан с одним пользователем («один») или не иметь такой связи («ноль»).

Связка «Заказ-Заказчик» на стороне заказа помечена «вороньей лапкой», то есть один и тот же заказчик может фигурировать в разных заказах, а на стороне заказчика связь помечена одной чертой, так как каждый заказ связан с одним конкретным заказчиком.

Представленная на рисунке 20 логическая модель базы данных включает 15 таблиц. Детали каждой таблицы будут раскрыты в описании физической модели данных. Физическое проектирование является завершающим этапом создания БД.

Логическая схема, созданная на этапе логического проектирования, детализируется с указанием всех типов, полей, размеров и ограничений. Сущности модели становятся таблицами, а их атрибуты столбцами [27].

На рисунке 21 представлена детализированная физическая модель базы данных, в которой указаны типы данных для каждого столбца. Первичные и внешние ключи имеют тип INT.

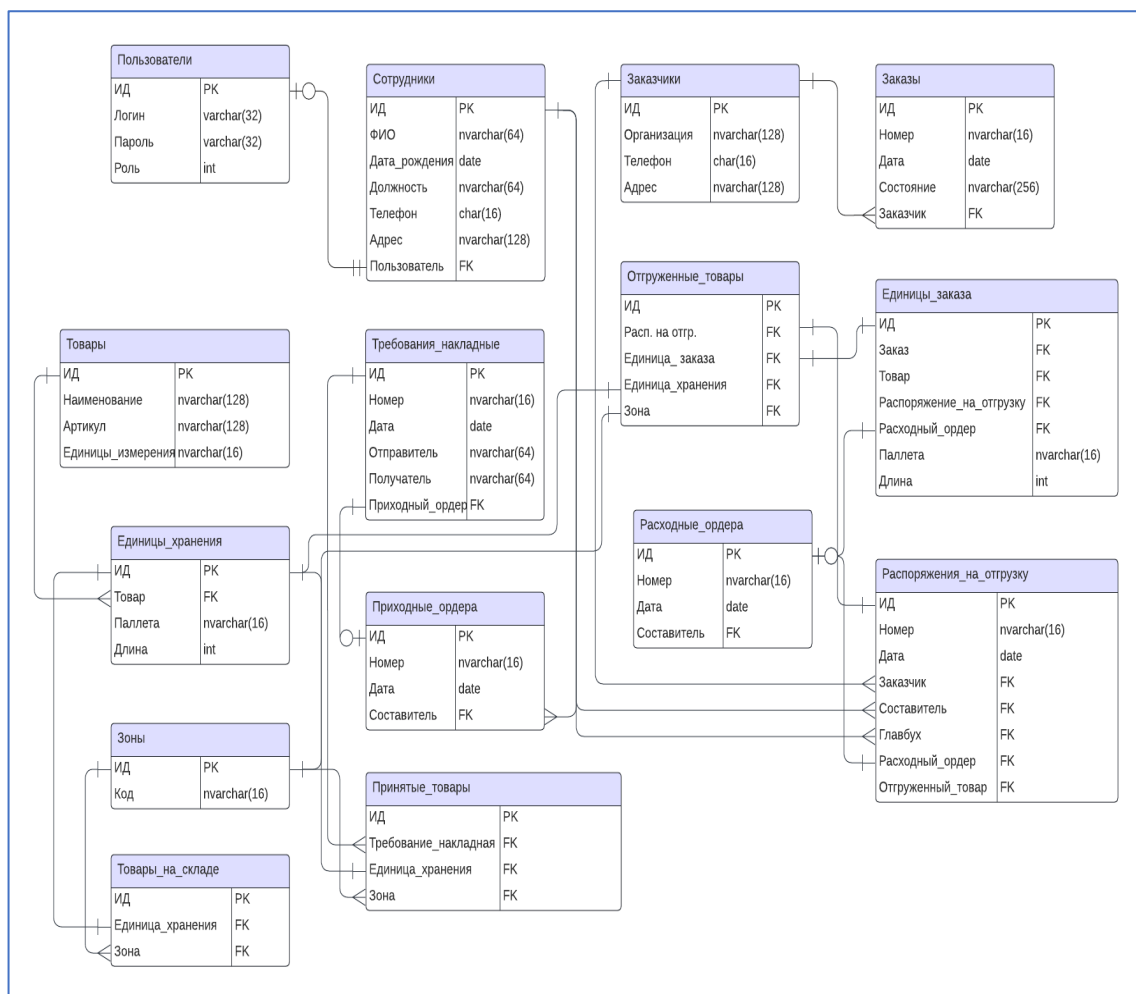


Рисунок 21 – Физическая модель базы данных

Структура таблиц БД иллюстрируется на рисунках 22-36.

DESKTOP-ON2VDDA... - dbo.Сотрудники			
	Имя столбца	Тип данных	Разрешить значения NULL
🔑	ИД	int	☐
	ФИО	nvarchar(64)	☐
	Дата_рождения	date	☐
	Должность	nvarchar(64)	☐
	Телефон	char(16)	☒
	Адрес	nvarchar(128)	☒
	Дата_принятия	date	☒
	Дата_увольнения	date	☒
	Пользователь	int	☒
	Полномочия	nvarchar(128)	☒

Рисунок 22 – Структура таблицы «Сотрудники»

Таблица «Сотрудники» содержит структуру хранения данных о сотрудниках, такие как ФИО, должность, контактные данные и другие относящиеся к персоналу сведения.

DESKTOP-ON2VDDA...use - dbo.Товары			
	Имя столбца	Тип данных	Разрешить значения NULL
🔑	ИД	int	☐
	Наименование	nvarchar(128)	☐
	Артикул	nvarchar(16)	☐
	Единицы_измерения	nvarchar(16)	☐
	Длина	int	☐
	Цена	money	☐
	Описание	ntext	☒

Рисунок 23 – Структура таблицы «Товары»

Таблица «Товары» содержит структуру хранения данных о товарах, включая наименование, артикул и другую аббревиатурную информацию, связанную с товарный номенклатурой.

DESKTOP-ON2VDDA...вары_на_отгрузку			
	Имя столбца	Тип данных	Разрешить значения NULL
🔑	ИД	int	☐
	Единица_заказа	int	☐
	Распоряжение_на_отгрузку	int	☐
	Единица_хранения	int	☒
	Зона	int	☒

Рисунок 24 – Структура таблицы «Товары_на_отгрузку»

Таблица «Товары_на_отгрузку» содержит записи о товарах, подготовленных для отгрузки.

DESKTOP-ON2VDDA...овары_на_складе			
	Имя столбца	Тип данных	Разрешить значения NULL
🔑	ИД	int	☐
	Единица_хранения	int	☐
	Зона	int	☐

Рисунок 25 – Структура таблицы «Товары_на_складе»

Таблица «Товары_на_складе» отражает информацию о текущих запасах товаров на складе, включая единицу измерения и зону хранения.

DESKTOP-ON2VDDA...о.Единицы_заказа			
	Имя столбца	Тип данных	Разрешить значения NULL
▶ 🔑	ИД	int	☐
	Заказ	int	☐
	Товар	int	☐
	Паллета	varchar(16)	☐
	Длина	int	☒
	Цена	money	☒
	Распоряжение_на_отгрузку	int	☒
	Расходный_ордер	int	☒

Рисунок 26 – Структура таблицы «Единицы_заказа»

Таблица «Единицы_заказа» содержит информацию заказа, наименование товара и другую аббревиатурную информацию, связанную с данными заказа.

DESKTOP-ON2VDD...диницы_хранения			
	Имя столбца	Тип данных	Разрешить значения NULL
🔑	ИД	int	☐
	Товар	int	☐
	Паллета	varchar(16)	☐
	Длина	int	☐
	Дата_поступления	date	☐
	Дата_отгрузки	date	☒

Рисунок 27 – Структура таблицы «Единицы_хранения»

Эта таблица предназначена для хранения информации о товаре, находящемся на складе, дате поступления и дате отгрузки данного товара.

DESKTOP-ON2VDDA\...- dbo.Заказчики			
	Имя столбца	Тип данных	Разрешить значения NULL
🔑	ИД	int	☐
	Организация	nvarchar(128)	☐
	Адрес	nvarchar(128)	☒
	Телефон	char(16)	☒

Рисунок 28 – Структура таблицы «Заказчики»

Таблица «Заказчики» хранит основные данные о клиентах и их контактных данных.

DESKTOP-ON2VDDA\...use - dbo.Заказы			
	Имя столбца	Тип данных	Разрешить значения NULL
🔑	ИД	int	☐
	Дата	date	☐
	Номер	nvarchar(16)	☐
	Заказчик	int	☐
	Состояние	nvarchar(256)	☒

Рисунок 29 – Структура таблицы «Заказы»

Таблица «Заказы» содержит данные о размещенных заказах, заказчике и статусе выполнения.

DESKTOP-ON2VDD\...овары_на_приеме			
	Имя столбца	Тип данных	Разрешить значения NULL
🔑	ИД	int	☐
	Единица_хранения	int	☐
	Требование_накладная	int	☐
	Зона	int	☒

Рисунок 30 – Структура таблицы «Товары_на_приеме»

Таблица «Товары_на_приеме» хранит информацию о товарах на этапе приемки, такие как данные документа требование-накладная, единица хранения и зона ответственного хранения.

Имя столбца	Тип данных	Разрешить значения NULL
ИД	int	☐
Код	nvarchar(16)	☐

Рисунок 31 – Структура таблицы «Зоны»

Таблица «Зоны» содержит данные зон хранения продукции.

Имя столбца	Тип данных	Разрешить значения NULL
ИД	int	☐
Логин	varchar(32)	☐
Пароль	varchar(32)	☐
Роль	int	☐

Рисунок 32 – Структура таблицы «Пользователи»

Таблица «Пользователи» содержит информацию о пользователях системы, включая роль и права доступа.

Имя столбца	Тип данных	Разрешить значения NULL
ИД	int	☐
Номер	nvarchar(16)	☐
Дата	date	☐
Оформил	nvarchar(32)	☐
Получил	nvarchar(32)	☒

Рисунок 33 – Структура таблицы «Приходные_ордера»

Таблица «Приходные_ордера» хранит информацию прихода товара, номер, дату, кто оформил и кто получил.

	Имя столбца	Тип данных	Разрешить значения NULL
🔑	ИД	int	☐
	Номер	nvarchar(16)	☐
	Заказчик	int	☐
	Дата	date	☐
	Разрешил	int	☐
	Главбух	int	☐
	Расходный_ордер	int	☒

Рисунок 34 – Структура таблицы «Распоряжения_на_отгрузку»

Таблица «Распоряжения_на_отгрузку» содержит информацию определяющие условия отгрузки, заказчика, даты, кто разрешил, главного бухгалтера.

	Имя столбца	Тип данных	Разрешить значения NULL
🔑	ИД	int	☐
	Номер	nvarchar(16)	☐
	Организация	nvarchar(64)	☐
	Дата_составления	date	☐
	Отправитель	nvarchar(64)	☐
	Получатель	nvarchar(64)	☐
	Затребовал	nvarchar(32)	☐
	Разрешил	nvarchar(32)	☐
	Приходный_ордер	int	☒

Рисунок 35 – Структура таблицы «Требования_накладные»

Таблица «Требования_накладные» содержит детальную информацию основного документа передачи продукции на склад из цеха.

DESKTOP-ON2VDDA...асходные_ордера			
	Имя столбца	Тип данных	Разрешить значения NULL
🔑	ИД	int	☐
	Номер	nvarchar(16)	☐
	Дата	date	☐
	Оформил	nvarchar(32)	☐

Рисунок 36 – Структура таблицы «Расходные_ордера»

Таблица «Расходные_ордера» содержит данные о расходе товара, номера, даты и кто оформил.

2.4 Требования к аппаратно-программному обеспечению ИС

Информационная система складского учета для ООО «Энерготехмаш» предполагает работу на операционной системе Windows. Рабочие станции, предназначенные для работы в системе, должны быть объединены в локальную сеть [4].

Для развертывания базы данных необходим отдельный сервер. Схема компонентов между рабочими станциями и сервером базы данных изображена на рисунке 37.

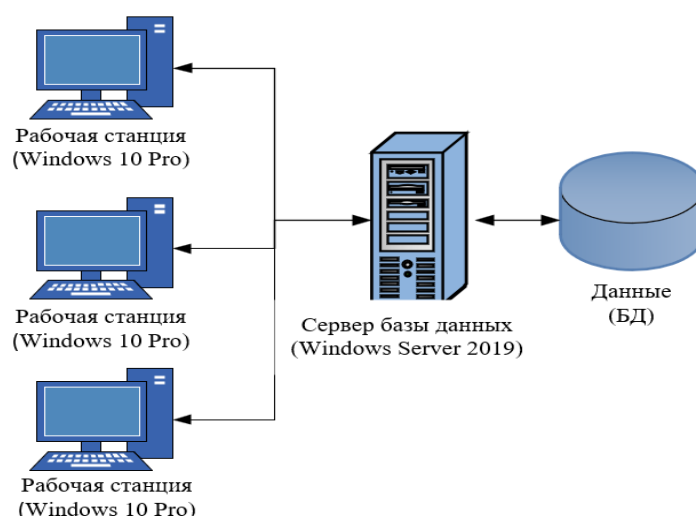


Рисунок 37 – Схема компонентов между рабочими станциями

Здесь возможны варианты, со своими достоинствами и недостатками. Можно установить собственный физический сервер в локальной сети.

Достоинствами такого подхода является полный контроль над сервером и независимость от подключения к Интернет. К недостаткам можно отнести необходимость приобретения сервера и несколько более сложное обслуживание физического сервера по сравнению с виртуальным [19].

Альтернативной является аренда виртуального сервера на каком-либо PaaS-сервисе. Достоинством является отсутствие единовременных затрат на приобретение сервера и несколько более простая процедура обслуживания сервера. Недостатками такого решения является зависимость от подключения к Интернет, ежемесячная плата за аренду сервера, а также отсутствие полного контроля над данными (например, PaaS-сервис может в любой момент ограничить доступ резидентам из России в связи с санкционным режимом, введенным западными странами против нашей страны). Ввиду последнего обстоятельства, первый вариант с приобретением собственного физического сервера кажется более предпочтительным.

Рекомендуемые системные требования для рабочих станций:

- процессор класса Intel Core i3 или выше;
- 8 ГБ оперативной памяти;
- накопитель 256 ГБ HDD/SSD;
- операционная система Windows 10 Pro или выше;
- сетевая карта с поддержкой Gigabit Ethernet;
- монитор с разрешением FullHD (1920x1080) или выше;
- периферийные устройства, такие как мышь, клавиатура и принтер (для печати документов) [17].

Рекомендуемые системные требования для сервера базы данных:

- процессор класса Intel Core i5 или выше;
- 16 ГБ или больше оперативной памяти;
- накопитель 512 ГБ SSD или больше;
- операционная система Windows Server 2019 или выше;

- система управления базами данных (СУБД) Microsoft SQL Server 2019 или выше (кроме Express версии).

Эти требования обеспечат комфортную и эффективную повседневную работу пользователей с системой складского учета.

Выводы по главе 2.

Во главе 2 на этапе логического проектирования была разработана логическая модель информационной системы, построены диаграмма вариантов использования системы и диаграмма классов, описывающая сущности системы и взаимоотношения между ними. Далее диаграмма классов была преобразована в ER-диаграмму базы данных. Подробно рассмотрено информационное обеспечение и требования к аппаратно-техническому обеспечению.

Глава 3 Физическое проектирование ИС

3.1 Критерии выбора архитектуры и технологии ИС

Из трёх известных архитектурных решений доступа к БД (файл-сервер, клиент-сервер, распределенная система), наиболее предпочтительной для разрабатываемой ИС кажется архитектура клиента-сервер [14].

Файл-серверная архитектура морально устаревшая и имеет слишком много недостатков. Распределенная система является избыточной для такого небольшого предприятия, которым является склад готовой продукции ООО «Энерготехмаш».

Итак, в разрабатываемой ИС будет использоваться клиент-серверная архитектура, которая может быть реализована с использованием различных моделей распределения функций между клиентом и сервером (рисунок 38).

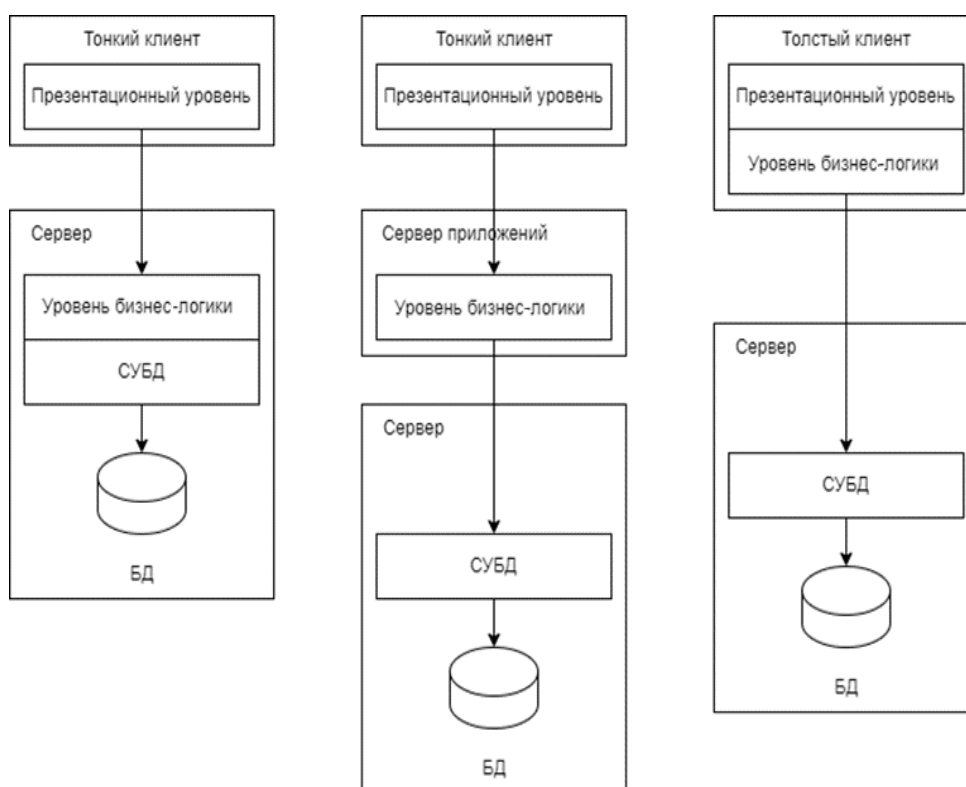


Рисунок 38 – Модели распределения функций между сервером и клиентом

В архитектуре «тонкого» клиента (thin client) клиентское приложение минимально сложное и предназначено только для ввода и вывода данных [18].

Данная модель имеет две разновидности – двухзвенную (бизнес-логика реализована на том же сервере, где находится БД) и трехзвенную (бизнес-логика реализована в отдельном сервере приложений) [21].

В модели с «толстым» клиентом (fat client) бизнес-логика реализована в клиенте. Каждая из моделей имеет свои достоинства и недостатки.

В модели с «тонким» клиентом значительно упрощается разработка клиентского приложения, которое, к тому же, может быть разработано с использованием различных технологий, например, одна версия клиента может быть реализована как десктопное приложение для смартфонов и так далее.

С другой стороны, требуется разработка двух отдельных приложений, клиента и сервера, что несколько увеличивает сложность системы.

В двухзвенной архитектуре вся вычислительная нагрузка ложится на сервер, поэтому может потребоваться относительно более мощный сервер. В трехзвенной архитектуре требуется наличие ещё одного сервера приложений, что является чрезмерным для небольшого предприятия.

«Толстый» клиент хоть и сложнее в разработке, чем «тонкий», но зато вся бизнес-логика реализована в нем самом, что снижает общую сложность разработки системы [22].

Так как основные вычисления выполняются на клиентских машинах, то значительно снижается нагрузка на сервер, и это позволяет использовать в качестве сервера гораздо менее производительную машину, чем потребовалось бы в системе с «тонким» клиентом [26].

Исходя из всего вышеизложенного, для разрабатываемой ИС выбрана система с «толстым» клиентом, так как подходит для такого небольшого предприятия, которым является склад готовой продукции ООО «Энерготехмаш» и упрощает требования к серверной части, снижая затраты на ее оборудование, обеспечивая тем самым сбалансированное решение с точки зрения экономичности, эффективности и простоты внедрения.

В качестве языка разработки ИС можно рассмотреть несколько альтернатив – C#, Java C++ или Python (два последних в связке с Qt для разработки графического интерфейса). Каждый из них пригоден для решения поставленной задачи и имеет мощные инструменты разработки.

Исходя из этого, выбран язык программирования C#, имеющий в своем составе такой мощный инструмент, как Entity Framework, который значительно упрощает разработчику работу с базой данных. Кроме того, в качестве СУБД будет выбрана MS SQL, продукт того же производителя, что должно положительно сказаться на совместимости и уменьшить количество ошибок при разработке ИС. Для разработки приложения выбрана интегрированная среда разработки IDE MS Visual Studio 2022. При проектировании информационной системы (ИС) планируется использование реляционной базы данных.

Среди самых популярных баз данных первые четыре места занимают Oracle, MySQL, Microsoft SQL Server и PostgreSQL. Oracle стоит немалых денег и ее использование не имеет смысла для малого предприятия. Остальные можно использовать для решения поставленной задачи [7].

Так как для разработки будет использоваться язык C# и IDE MS Visual Studio, то наиболее предпочтительным выбором очевидно использование Microsoft SQL Server, что гарантирует максимальную совместимость приложения и БД и имеет ряд преимуществ:

- современные средства разработки (удобный интерфейс, CASE-инструменты, триггеры, наличие средств отладки);
- гарантирует надежность и защиту данных (авторизация, аутентификация);
- инструментарий администратора (резервное копирование и восстановление, контроль производительности, введение протокола работы СУБД, импорт/экспорт данных) [3].

3.2 Разработка ИС учета товаров на складе

При разработке приложений баз данных с использованием C# и Entity Framework (EF) для отображения таблиц базы данных на классы объектно-ориентированной модели информационной системы используем подход «Database First», предполагающий создание таблиц вручную. Для этого предназначено приложение MS SQL Server Management Studio, имеющее интуитивно понятный интерфейс. После подключения к серверу БД была создана база данных «Energotehmash-Warehouse» и добавлены все таблицы, поля и связи, в соответствии с разработанной ER-диаграммой (рисунок 20). Схема модулей приложения показана на рисунке 39.

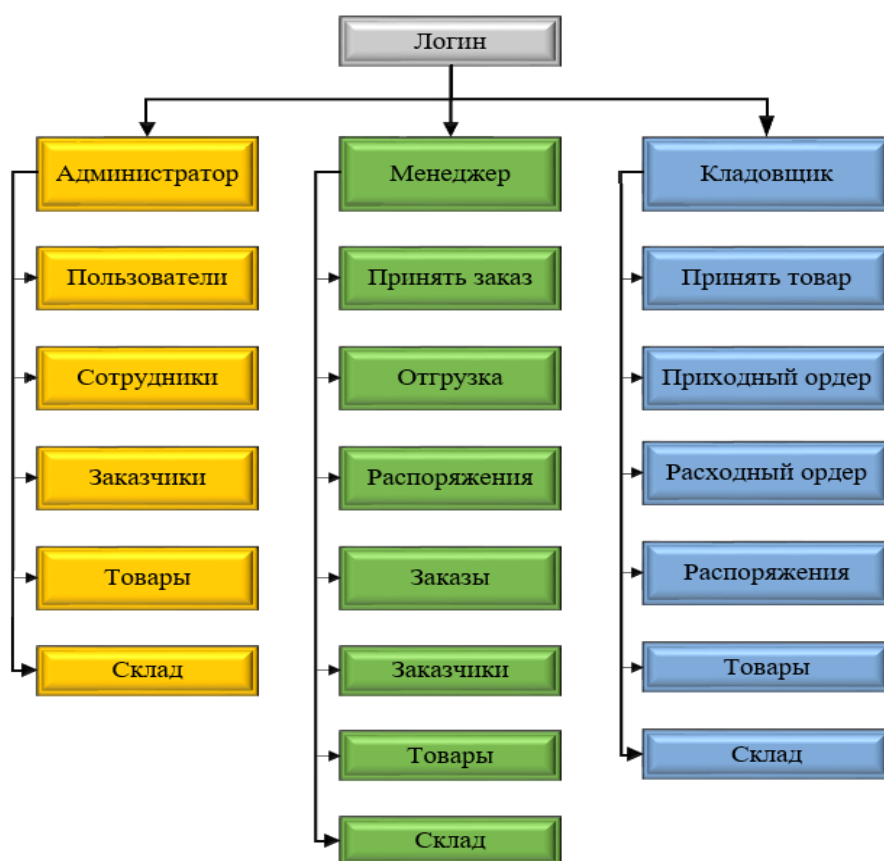


Рисунок 39 – Схема модулей приложения

Приложение имеет модульную структуру, с тремя главными модулями трех ролей пользователей, и несколькими субмодулями, некоторые из которых доступны для любых пользователей, а некоторые только для определенной роли.

Проект приложения в MS Visual Studio состоит из 21 основного файла с программным кодом.

В таблице 4 даны названия файлов и их краткое описание.

Таблица 4 – Описание файлов проекта

Файл	Описание
AcceptOrderForm	Принять заказ
AcceptProductsForm	Принять товар
AdminForm	Главное окно «Администратора»
CustomersForm	Заказчики
EmployeesForm	Сотрудники
GoodsForm	Товары (добавление, печать отчетов «Карточка учета товара» и «Ведомость по товарам на складе»)
InOrderForm	Приходный ордер (создать новый, посмотреть старый, распечатать любой)
InOrderListForm	Список всех приходных ордеров
LoginForm	Форма для ввода логина/пароля
MainForm	Главное окно приложения
ManagerForm	Главное окно «Менеджера»
OrderForm	Заказы
OutOrderForm	Расходный ордер (создать новый, посмотреть старый, распечатать любой)
OutOrderListForm	Список всех расходных ордеров
ShippingOrderForm	Распоряжение на отгрузку (создание)
ShippingOrderListForm	Список распоряжений на отгрузку
ShippingOrderViewForm	Распоряжение на отгрузку (просмотр)
StorageForm	Склад. Товары на хранении
StorekeeperForm	Главное окно «Кладовщика»
UsersForm	Пользователи

Схема меню модуля приложения «Администратор» представлена на рисунке 40.

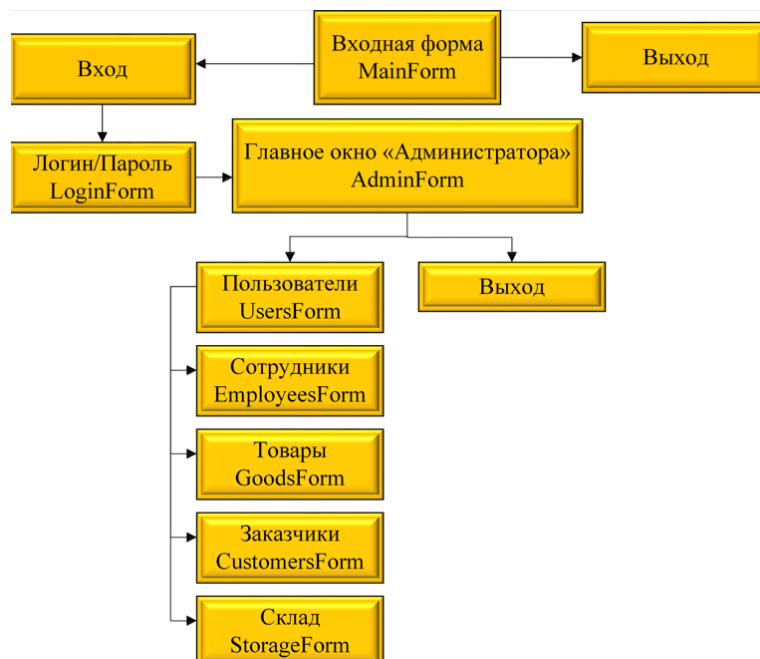


Рисунок 40 – Схема меню модуля приложения «Администратор»

Модуль «Пользователи» (UsersForm) имеет функцию управления пользователями системы. Позволяет добавить или удалить пользователя, изменить роль, логин или пароль. Файлы в виде конструктора и программного кода модуля представлена на рисунке 41-42.

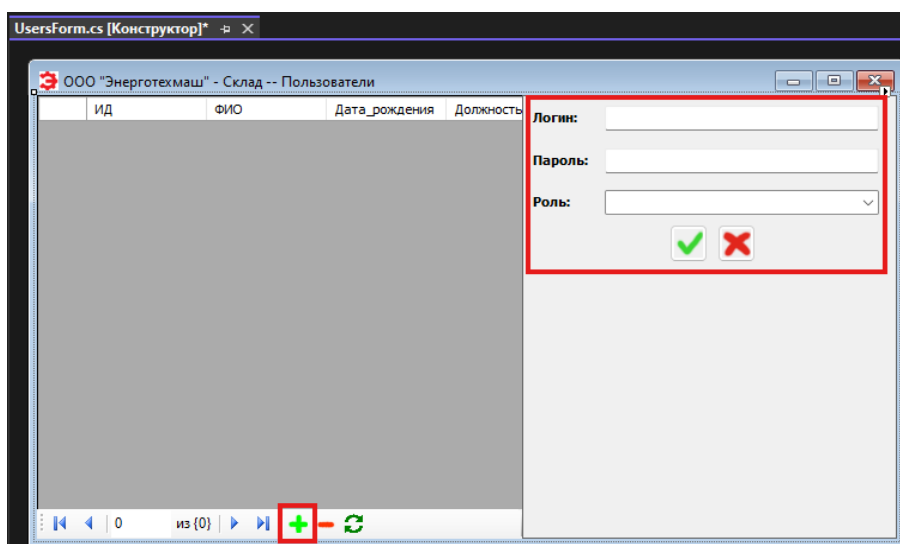


Рисунок 41 – UsersForm в виде конструктора

Все поля в данной форме должны быть заполнены. При нажатии кнопки плюс происходит добавление пользователя в систему.

```
if (!checkFields(ID < 0))
{
    MessageBox.Show("Не все поля заполнены!");
    return;
}

if (ID > 0)
{
    try
    {
        Пользователи user = context.Пользователи.Where(x => x.ИД == ID).Single();
        user.Пароль = textBoxPassword.Text;
        user.Роль = (int)comboBoxRole.SelectedValue;
        context.SaveChanges();
        fill();
    }
    catch (Exception)
    {
        MessageBox.Show("Ошибка при изменении пользователя.");
    }
}
```

Рисунок 42 – Фрагмент кода UsersForm

Данный фрагмент кода проверяет заполненность полей, выводит сообщение об ошибке, если не все поля заполнены.

Схема меню модуля приложения «Кладовщик» представлена на рисунке 43.



Рисунок 43 – Схема меню модуля приложения «Кладовщик»

Каждый из предоставленных модулей обладает функциями управления складскими операциями, добавить или удалить, просмотр, формирование документов и печать. Модуль «Принять товар» (AcceptProductsForm) представлен в виде конструктора и программного кода на рисунках 44-45.

Рисунок 44 – AcceptProductsForm в виде конструктора

Все поля в данной форме должны быть заполнены. При нажатии кнопки «Принять» добавляется информация о товаре в систему.

```
// Создать [Требование_накладная]
Требования_накладные invoice = new Требования_накладные
{
    Номер = textBoxNumber.Text,
    Организация = textBoxOrg.Text,
    Дата_составления = dateTimePicker1.Value.Date,
    Отправитель = textBoxSender.Text,
    Получатель = textBoxReceiver.Text,
    Затребовал = textBoxRequester.Text,
    Разрешил = textBoxResolver.Text,
    Приходный_ордер = null
};

// Создать [Единицы_хранения] и [Товары_на_приеме]
foreach (DataGridViewRow row in dataGridView1.Rows)
{
    Товары product = row.Tag as Товары;
    Единицы_хранения unit = new Единицы_хранения
    {
        Товар = product.ИД,
        Дата_поступления = dateTimePicker1.Value.Date,
        Паллета = ((Pallet)row.Cells[3].Value).ToString(),
        Длина = (int)row.Cells[4].Value
    };
    context.Единицы_хранения.Add(unit);

    Товары_на_приеме inProduct = new Товары_на_приеме
    {
        Требования_накладные = invoice,
        Единицы_хранения = unit
    };
    context.Товары_на_приеме.Add(inProduct);
}
```

Рисунок 45 – Фрагмент кода AcceptProductsForm

При нажатии кнопки «Принять», данный фрагмент кода переносит информацию, заполненную пользователем, в базу данных.

Схема меню модуля приложения «Менеджер» представлена на рисунке 46.

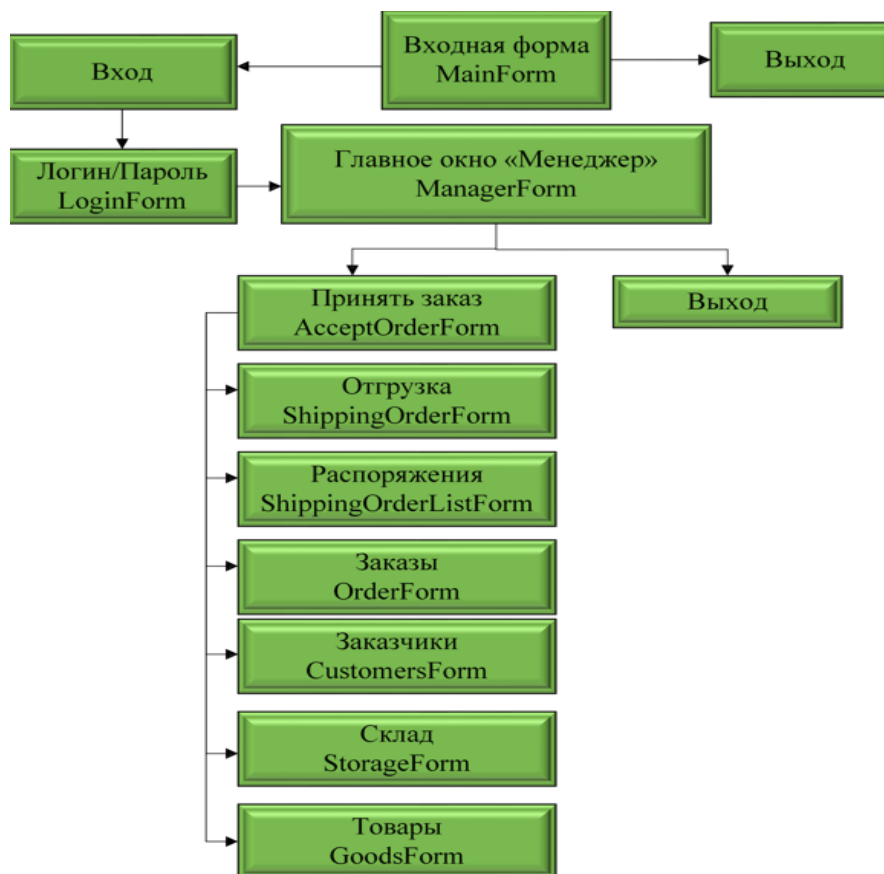


Рисунок 46 – Схема меню модуля приложения «Менеджер»

Каждый из предоставленных модулей обладает функциями управления операций, связанных с заказом товара, добавление или удаление, просмотр, формирование документов и печать.

Файлы модуля «Отгрузка» (ShippingOrderForm) в виде конструктора и программного кода модуля представлена на рисунке 47-48.

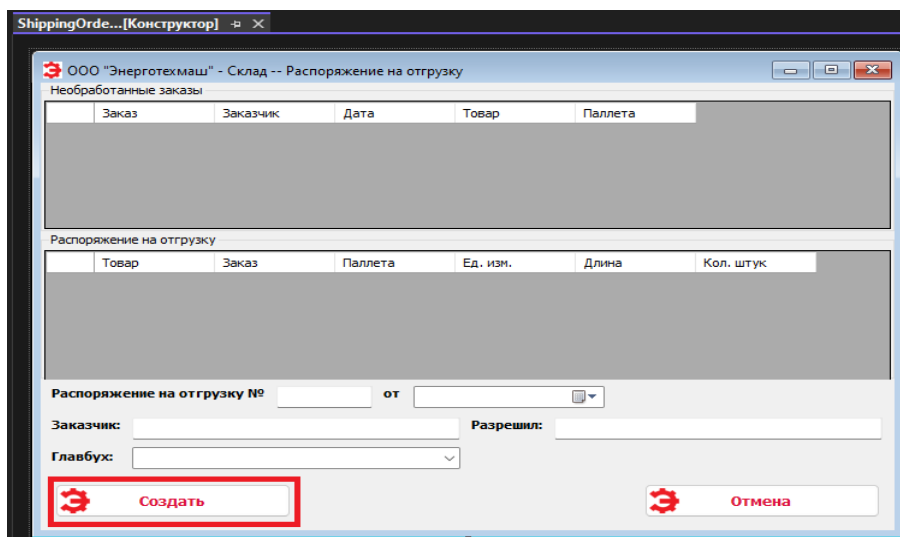


Рисунок 47 – ShippingOrderForm в виде конструктора

При нажатии кнопки «Создать», создается распоряжение на отгрузку и присваивается заказу состояние «Ожидает отгрузки».

```
private void buttonAccept_Click(object sender, EventArgs e)
{
    if (!created)
    {
        if (!checkOrder()) { return; }

        shipOrder = new Распоряжения_на_отгрузку
        {
            Номер = textBoxNumber.Text,
            Дата = dateTimePicker1.Value.Date,
            Разрешил = manager.ИД,
            Главбух = (int)comboBoxBooker.SelectedValue
        };

        foreach (DataGridViewRow row in dataGridView2.Rows)
        {
            OrderItem item = row.Tag as OrderItem;
            shipOrder.Заказчик = item.customerId;
            Единицы_заказа unit = context.Единицы_заказа.Where(x => x.ИД == item.unitId).Single();
            shipOrder.Единицы_заказа.Add(unit);

            Товары_на_отгрузку outProd = new Товары_на_отгрузку();
            outProd.Единицы_заказа = unit;
            outProd.Распоряжения_на_отгрузку = shipOrder;
            outProd.Единицы_хранения = null;
            outProd.Зона = null;

            shipOrder.Товары_на_отгрузку.Add(outProd);
        }

        context.Распоряжения_на_отгрузку.Add(shipOrder);
        context.SaveChanges();

        // Обновление состояния связанных заказов
        // если для всех единиц заказа было выписано распоряжение
        foreach (DataGridViewRow row in dataGridView2.Rows)
        {
            OrderItem item = row.Tag as OrderItem;
            Заказы order = context.Заказы.Where(x => x.ИД == item.orderId).Single();

            if (order.Состояние != "Ожидает отгрузки")
```

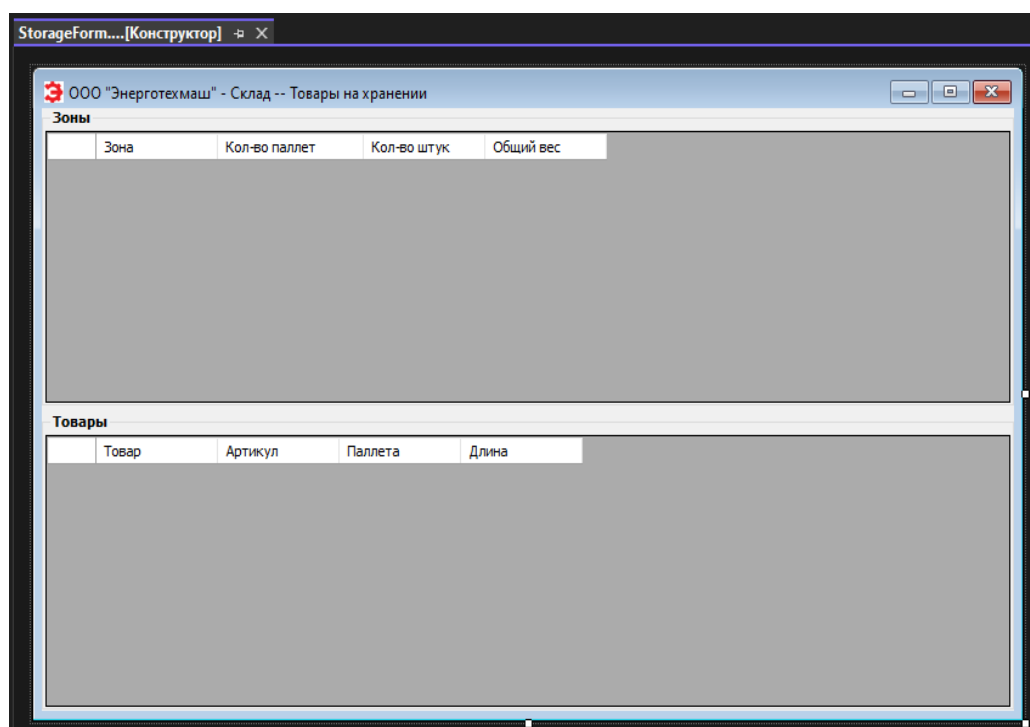
Рисунок 48 – Фрагмент кода ShippingOrderForm

При нажатии кнопки «Создать», данный фрагмент кода создает распоряжение на отгрузку и присваивает заказу состояние «Ожидает отгрузки».

Отметим, что такие модули как «Склад» и «Товары» присутствуют во всех модулях и доступны для всех пользователей.

Модуль «Склад» позволяет получить информацию о занятости всех зон склада с целью, во-первых, найти свободное место для размещения товаров, а во-вторых, получить информацию о полной заполненности склада.

Файлы в виде конструктора и программного кода модуля «Склад» (StorageForm) представлены на рисунках 49-50.



Рисунке 49 – StorageForm в виде конструктора

В верхней части отражается занятость зоны склада, а в нижней части формы товары, находящиеся на складе.

```

private void dataGridview1_SelectionChanged(object sender, EventArgs e)
{
    var row = dataGridview1.CurrentRow;
    if (row != null)
    {
        dataGridview2.Rows.Clear();

        string code = (string)row.Cells[0].Value;
        if (code != null)
        {
            Зоны zone = context.Зоны.Where(x => x.Код == code).Single();
            foreach (var rel in zone.Товары_на_складе)
            {
                int unitId = rel.Единица_хранения;
                Единицы_хранения unit = context.Единицы_хранения.Where(x => x.ИД == unitId).Single();
                Товары product = context.Товары.Where(x => x.ИД == unit.Товар).Single();
                int i = dataGridview2.Rows.Add();
                var row2 = dataGridview2.Rows[i];
                row2.Cells[0].Value = product.Наименование;
                row2.Cells[1].Value = product.Артикул;
                row2.Cells[2].Value = unit.Паллета;
                row2.Cells[3].Value = unit.Длина;
            }
        }
    }
}

```

Рисунок 50 – Фрагмент кода файла StorageForm

Данный фрагмент кода выводит информацию о товарах, находящихся на складе.

Модуль «Товары» позволяет видеть список всех товаров и их количество с функцией только добавления нового товара, формирования и печати отчетов «Карточка учета товара» и «Ведомость по товарам на складе», путем нажатия на соответствующие значки на нижней панели макета (рисунок 52).

Файлы в виде конструктора и программного кода модуля «Товары» (GoodsForm) представлены на рисунках 51-52.

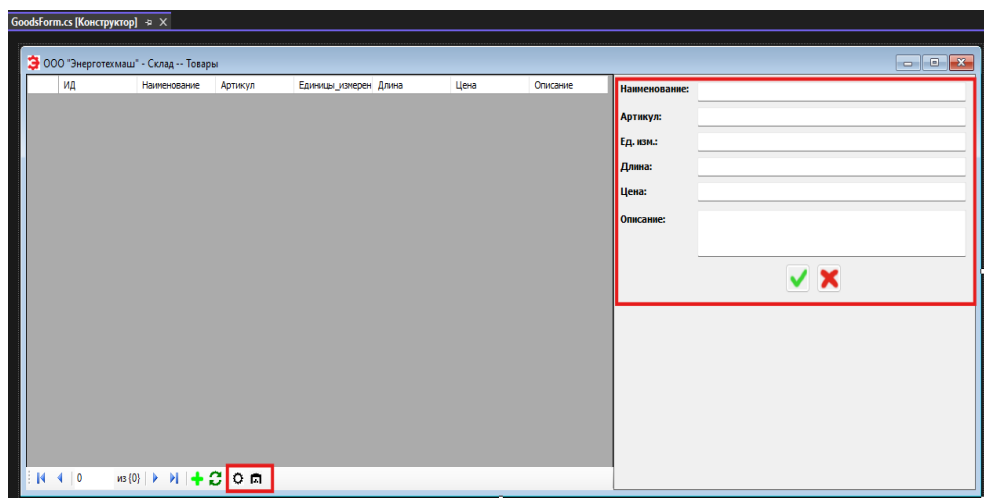


Рисунок 51 – GoodsForm в виде конструктора

Все поля должны быть заполнены и при нажатии на знак галочка добавляется информация о товаре в систему.

```
private void buttonOk_Click(object sender, EventArgs e)
{
    var row = dataGridViewGoods.CurrentRow;
    if (row != null)
    {
        int ID = (int)row.Cells[0].Value;
        if (!checkFields(ID < 0))
        {
            MessageBox.Show("Не все поля заполнены!");
            return;
        }

        if (ID > 0)
        {
            try
            {
                Товары product = context.Товары.Where(x => x.ИД == ID).Single();
                product.Наименование = textBoxName.Text;
                product.Артикул = textBoxCode.Text;
                product.Единицы_измерения = textBoxUnits.Text;
                product.Длина = Int32.Parse(textBoxLength.Text);
                product.Цена = decimal.Parse(textBoxPrice.Text);
                product.Описание = textBoxInfo.Text;
                context.SaveChanges();
                fill();
            }
            catch (Exception)
            {
                MessageBox.Show("Ошибка при изменении товара.");
            }
        }
        else
        {
            try
            {
                Товары product = new Товары
                {
                    Наименование = textBoxName.Text,
```

Рисунок 52 – Фрагмент кода файла GoodsForm

Данный фрагмент кода проверяет заполненность полей и добавляет информацию о товаре в базу данных.

Таким образом, разработка программного обеспечения для ИС успешно реализована с использованием C# и Entity Framework (EF), применяя подход «Database First». Созданное приложение имеет модульную структуру, охватывающая все основные процессы управления складом.

3.3 Функциональное тестирование ИС

Функциональное тестирование фокусируется на проверке того, как система выполняет свои заявленные функции на основе заданных требований [25].

Поскольку этот вид тестирования оценивает работу программы без знания ее внутренней структуры или кода, его относят к методам тестирования «черного ящика» [24].

Перед началом тестирования приложения необходимо (пользователем «Администратор») создать несколько сотрудников (три пользователя создаются при выполнении SQL-скрипта) – менеджера (с привязкой к пользователю с ролью «Менеджер»), кладовщика (с привязкой к пользователю с ролью «Кладовщик») и главного бухгалтера (без привязки к пользователю).

Также необходимо добавить хотя бы несколько видов товаров (модуль «Товары») и заказчиков (модуль «Заказчики»). После запуска приложения открывается главное окно с кнопками «Вход» и «Выход» (рисунок 53).

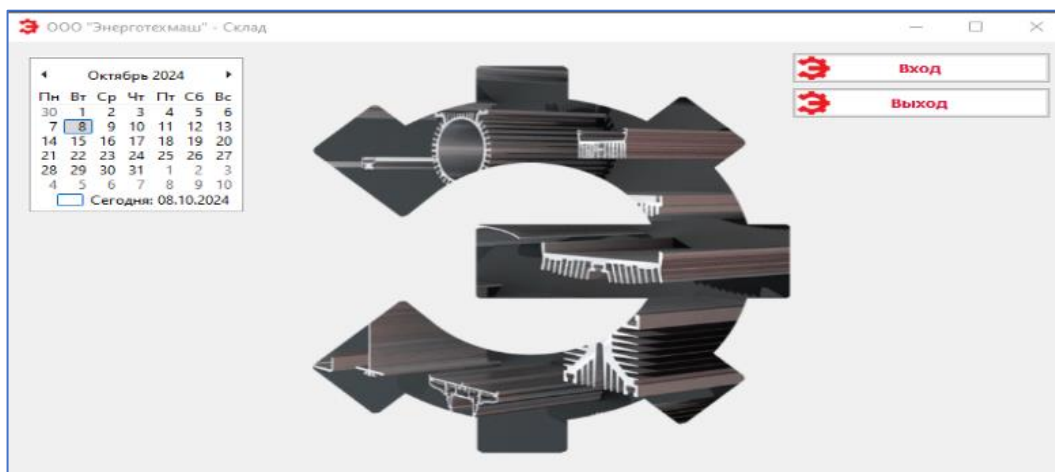


Рисунок 53 – Главное окно

Для входа необходимо нажать кнопку «Вход» и откроется окно для ввода логина и пароля (рисунок 54).

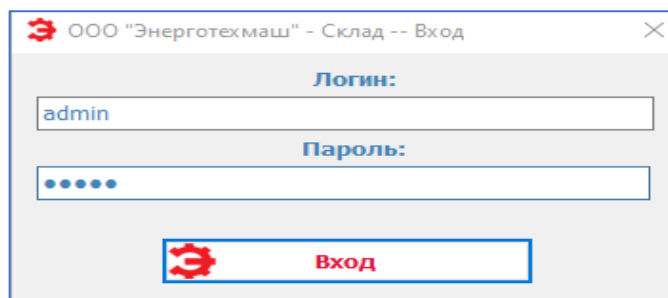


Рисунок 54 – Вход в систему

После успешной авторизации запустится один из трех модулей, в зависимости от роли пользователя.

На рисунке 55 показано главное окно модуля «Администратор».

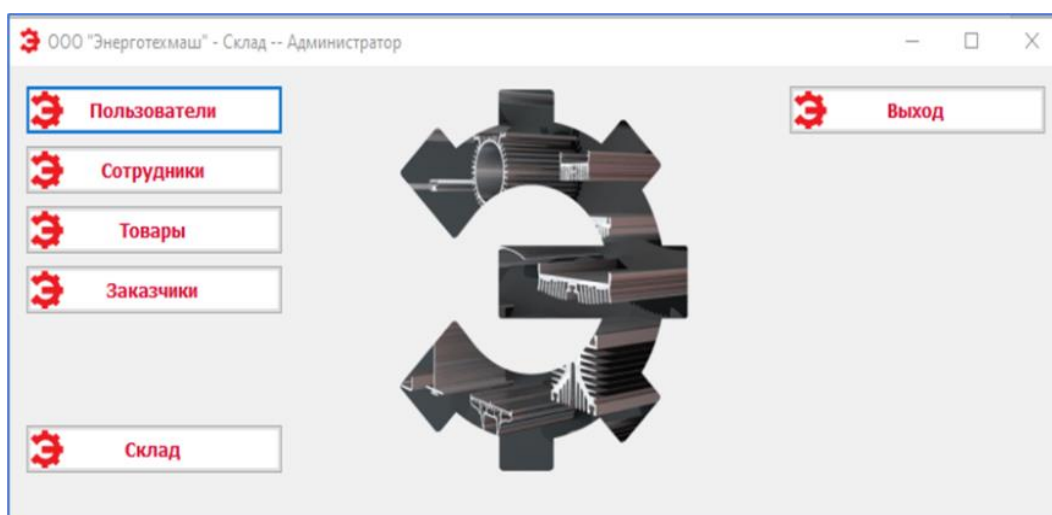


Рисунок 55 – Модуль «Администратор»

Субмодуль «Пользователи» позволяет посмотреть список всех пользователей, добавить нового пользователя, изменить пароль существующего пользователя или удалить существующего пользователя.

Изменение пароля пользователя происходит при выборе строки в таблице и изменение пароля в соответствующем поле в правой панели (рисунок 56).

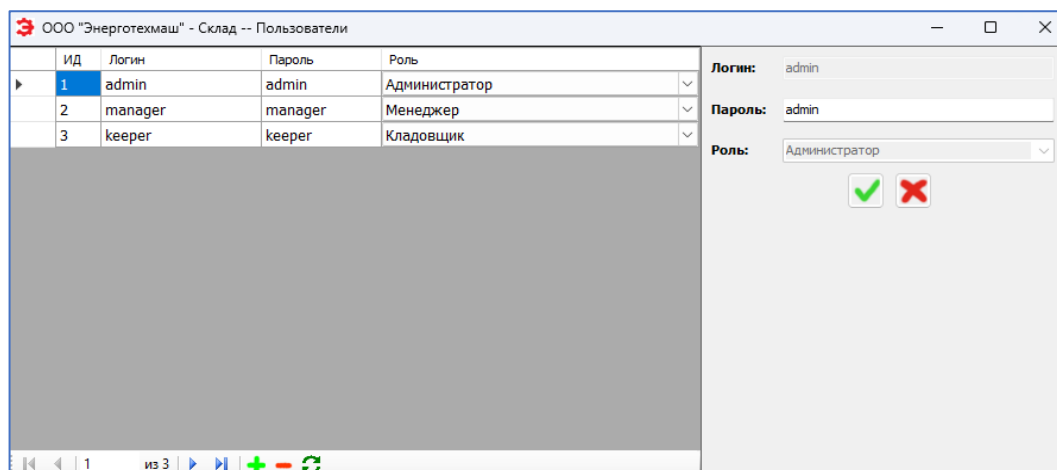


Рисунок 56 – Изменение пароля пользователя

Внесение изменений в базу данных произойдет после нажатия кнопки с зелёной галочкой. При нажатии с красным крестиком произойдет отмена введенных изменений. Добавление нового пользователя происходит после нажатия на кнопку с зеленым знаком плюс (рисунок 57).

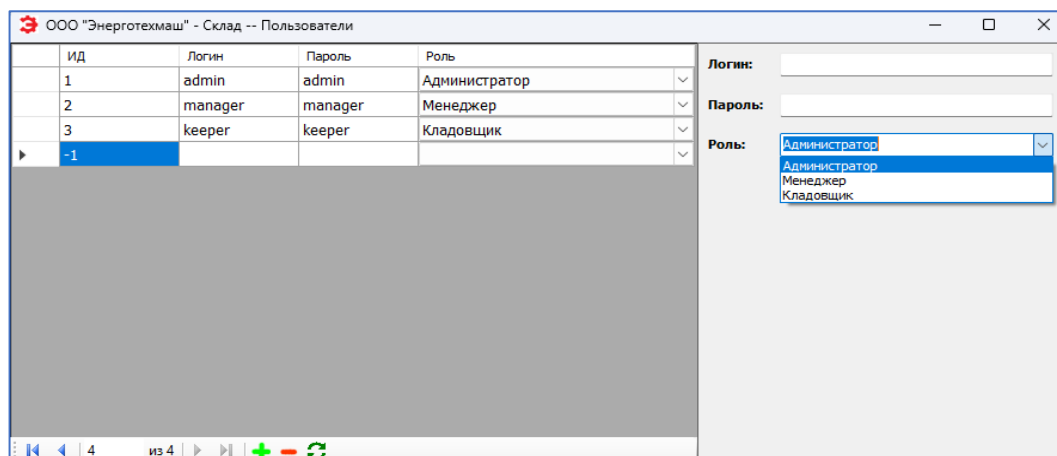


Рисунок 57 – Добавление нового пользователя

Аналогично, внесение изменений в базу данных произойдет после нажатия кнопки с галочкой (все поля должны быть заполнены, иначе появится сообщение). При нажатии кнопки с крестиком произойдет отмена добавления пользователя. Удаление пользователя происходит при выборе строки в

таблице и нажатии на кнопку с символом минус. Работа с остальными справочниками («Сотрудники», «Заказчики», «Товары») происходит аналогично. На рисунке 58 показана форма для работы с сотрудниками.

ИД	ФИО	Дата_рожде	Должность	Телефон	Адрес	Дата_принят	Дата_увольнения	Пользователь
1	Акимова Марина Владимировна	16.06.1999	Кладовщик	89967345754	г.Жигулевск, ул. Бо...	08.06.2023		keereg
2	Саушкина Ирина Витальевна	03.09.1998	Менеджер	83456732789	г. Тольятти, ул. Юб...	09.08.2022		manager
3	Егоров Константин Олегович	25.02.2000	Администратор	88745689022	г. Жигулевск, ул. Ку...	08.06.2023		admin

ФИО: Акимова Марина Владимировна
Дата рождения: 16 июня 1999 г.
Должность: Кладовщик
Телефон: 89967345754
Адрес: г.Жигулевск, ул. Бородинская, д.4 кв.15
Дата принятия: 8 июня 2023 г.
Полномочия: Кладовщик склада готовой продукции
Пользователь: keereg

Рисунок 58 – Модуль «Сотрудники»

На рисунке 59 показана форма для работы с товарами.

ИД	Наименование	Артикул	Единицы измерения	Длина	Цена	Описание
1	ПП-1442 БП	000001	кг	2600	180,0000	Профиль экструдированный, без ...
2	ПП-1397 БП	000002	кг	2550	195,0000	Профиль экструдированный

Наименование: ПП-1442 БП
Артикул: 000001
Ед. изм.: кг
Длина: 2600
Цена: 180,0000
Описание: Профиль экструдированный, без покрытия

Рисунок 59 – Модуль «Товары»

Две крайние справа кнопки в нижней панели навигации предназначены для печати отчетов «Карточка учета товара» (рисунок 60) и «Ведомость по товарам на складе».

Типовая межотраслевая форма № М-17
Утверждена постановлением Госкомстата России
от 30.10.97 № 71а

КАРТОЧКА № 1
учета материалов

Организация ООО «Энерготехмаш»
Структурное подразделение Склад готовой продукции

Форма по ОКУД
по ОКПО
Дата составления

Коды
0315008
65478124
9 10 2024

Структурное подразделение	Вид деятельности	Склад	Место хранения	Марка	Сорт	Профиль	Размер	Номенклатурный номер	Единица измерения	Цена, руб. коп.	Норма запаса	Срок годности	Поставщик
Склад готовой продукции				ПП-1442 БП				000001	кг	180,00			ООО «Энерготехмаш»

Наименование материала ПП-1442 БП Профиль экструдированный

Драгоценный материал (металл, камень)							
наименование	вид	номенклатурный номер	единица измерения	код	наименование	количество (масса)	номер паспорта
1	2	3	4	5	6	7	8

Дата записи	Номер		От кого получено или кому отпущено	Учетная единица выпуска продукции (работ, услуг)	Приход	Расход	Остаток	Подпись, дата
1	2	3						
09.10.24	1	1	ООО «Энерготехмаш», накладная № 10564 от 09.10.2024	кг	254		254	Саушкина 09.10.2024
09.10.24	1	2	Отпущено ООО «25 МИКРОН»	кг		254	0	Саушкина 09.10.2024

Карточку заполнил Менеджер Саушкина Саушкина Ирина Витальевна
(подпись) (подпись) (расшифровка подписи)
" 9 " октября 20 24 г.

Рисунок 60 – Карточка учета товаров

Ведомость по товарам на складе формирует отчет по всем видам товара за выбранный период (рисунок 61).

Ведомость по товарам на складе

Параметры: Период: 09.10.2024 - 09.10.2024

Количество товаров: В единицах хранения

Склад: Склад готовой продукции

Артикул	Товар	Ед. изм.	Начальный остаток	Приход	Расход	Конечный остаток
000001	ПП-1442 БП	кг	0	254	254	0
000002	ПП-1397 БП	кг	0	129	129	0

Рисунок 61 – Ведомость по товарам на складе

На рисунке 62 показана форма для добавления заказчиков. При нажатии кнопки с зеленой галочкой произойдет добавление заказчика, после заполнения полей с данными заказчика.

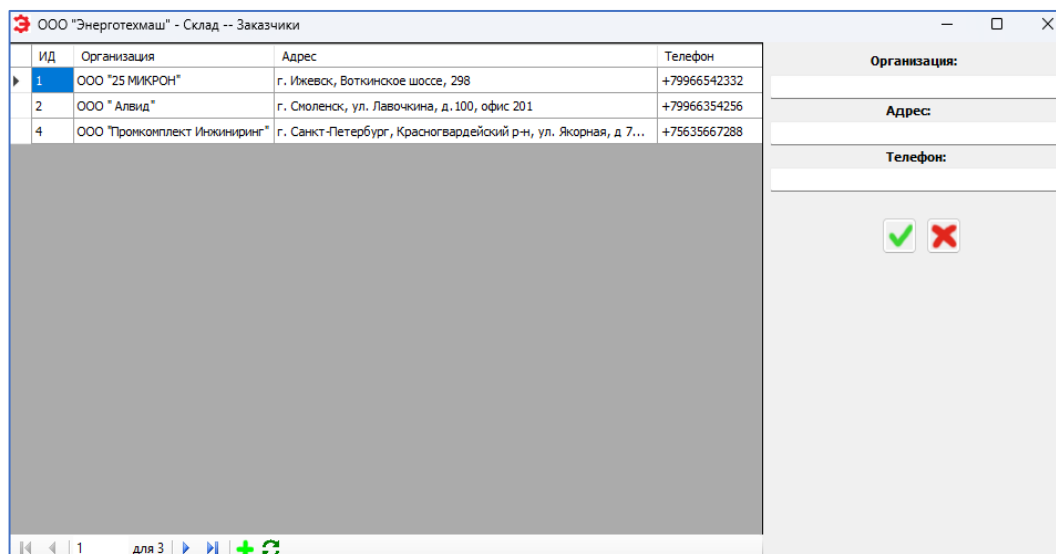


Рисунок 62 – Модуль «Заказчики»

На рисунке 63 показано главное окно модуля «Кладовщики» с ключевыми процессами функционального рабочего места.

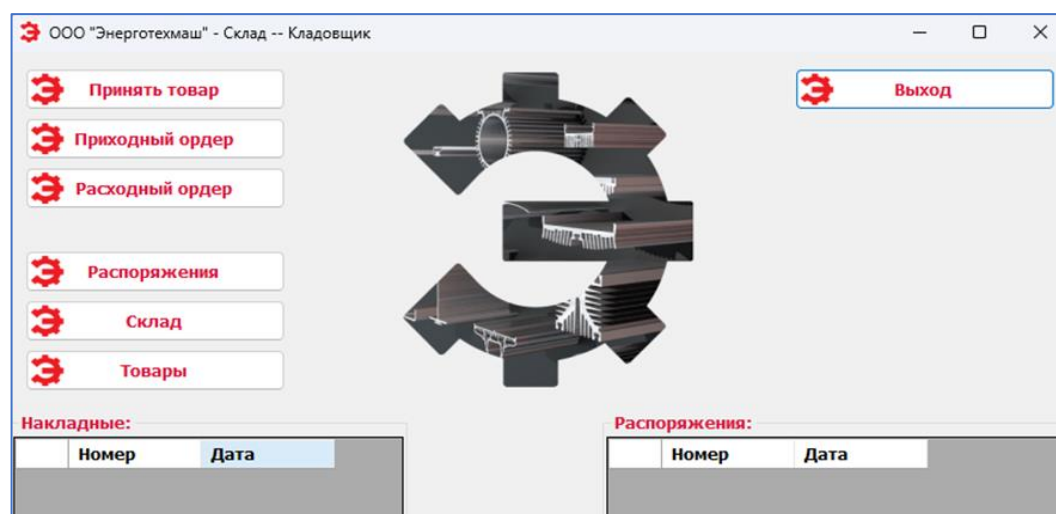


Рисунок 63 – Главное окно модуля «Кладовщик»

Внизу слева отображается список необработанных требований-накладных, справа – необработанные распоряжения на отгрузку.

При двойном щелчке на элементе списка слева появиться окно для оформления приходного ордера, а справа – расходного ордера.

Работа кладовщика начинается с приема товаров (рисунок 64).

ООО "Энерготехмаш" - Склад -- Принять товар

Требование-накладная

Номер накладной: 10564

Организация: ООО "Энерготехмаш" Дата составления: 9 октября 2024 г.

Отправитель: Цех №1 Получатель: Склад готовой продукции (СП)

Затребовал: Саушкина Ирина Витальевна Разрешил: Толкачев Виталий Сергеевич

Товар	Артикул	Паллета	Ед. изм.	Длина	Кол-во, шт
▶ ПП-1442 БП	000001	кг	1620шт, 254кг	2600	1620
ПП-1397 БП	000002	кг	257шт, 129кг	2550	257

Товар: ПП-1397 БП Паллета (шт/кг): 257 129 Длина, мм: 2550

Принять Отклонить

Рисунок 64 – Прием товаров

Необходимо ввести все данные из требования-накладной и заполнить список товаров.

Алгоритм добавления товаров, следующий: необходимо выбрать название товара из выпадающего списка (он содержит только товары из базы данных).

Далее нужно заполнить поле «Паллета». Количество и вес вводится в одном поле, в качестве разделителя можно использовать пробел, запятую, точку с запятой, дефис или слеш, а также можно ввести полностью, например «100шт,200кг». В последнем поле нужно ввести длину.

После заполнения этих полей нужно нажать кнопку со знаком плюс.
Если при вводе не произошло ошибки, то товар добавляется в список.

После ввода всех товаров нужно нажать кнопку «Принять».

Если форма заполнена без ошибок, то окно закроется, а в главном окне в левом списке появиться новый элемент – новая необработанная накладная.

При двойном щелчке по этой накладной появится форма для создания приходного ордера (рисунок 65).

ООО "Энерготехмаш" - Склад -- Приходный ордер

Приходный ордер № 1 от 9 октября 2024 г.

Отправитель: Цех №1

Получатель: Склад готовой продукции (СГП)

Распоряжение: Требование-накладная

Входящий документ: № 10564 от 9 октября 2024 г.

Оформил: Акимова Марина Владимировна

	№	Зона	Артикул	Товар	Паллета	Кол-во
▶	1	A	000001	ПП-1442 БП	1620шт, 254кг	1620
	2	A	000002	ПП-1397 БП	257шт, 129кг	257

Создать Отмена

Рисунок 65 – Приходный ордер

Все поля уже заполнены данными из требования-накладной (но номер ордера и дата свои, не из накладной).

Кладовщику нужно лишь распределить товар по зонам. Для этого для каждого товара необходимо выбрать букву зоны из выпадающего списка. После нажатия на кнопку «Создать» кнопки изменят свои названия и все элементы ввода станут неактивными – теперь данный ордер можно только распечатать (рисунок 66).

Приходный ордер № 1 от 9 октября 2024 г.

Отправитель: ООО "Энерготехмаш"

Склад: Склад готовой продукции

Распоряжение: Требование-накладная

Входящий документ: № 10564 от 9 октября 2024 г.

№ п/п	Зона	Артикул	Товар	Паллета	Ед. изм.	Длина, мм	Кол-во, шт
1	А	000001	ПП-1442 БП	1620шт,254кг	кг	2600	1620
2	Б	000002	ПП-1397 БП	257шт,129кг	кг	2550	257

Всего 2 наименований

Отпустил _____ Акимова Марина Владимировна
(подпись) (должность) (Ф.И.О.)

Получил _____
(подпись) (должность) (Ф.И.О.)

Рисунок 66 – Печатная форма приходного ордера

После создания приходного ордера, можно увидеть, что товары появились в модуле «Склад» (рисунок 67).

ООО "Энерготехмаш" - Склад -- Товары на хранении				
Зоны				
	Зона	Кол-во паллет	Кол-во штук	Общий вес
▶	А	1	1620	254
	Б	1	257	129
Товары				
	Товар	Артикул	Паллета	Длина
▶	ПП-1442 БП	000001	1620шт,254кг	2600

Рисунок 67 – Модуль «Склад»

В верхней таблице отображается каждая зона, в которой есть хотя бы одна единица хранения. В нижней таблице отображается список товаров в выбранной зоне.

На рисунке 68 показано главное окно модуля «Менеджер».

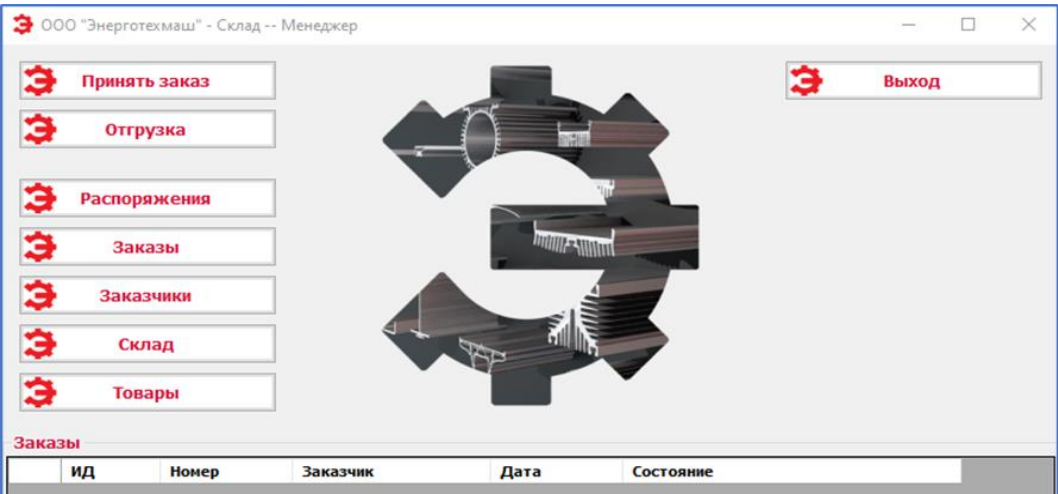


Рисунок 68 – Интерфейс модуля «Менеджер»

Работа менеджера начинается с приема заказов (рисунок 69), процесс которого во многом похож на прием товаров, описанный выше.

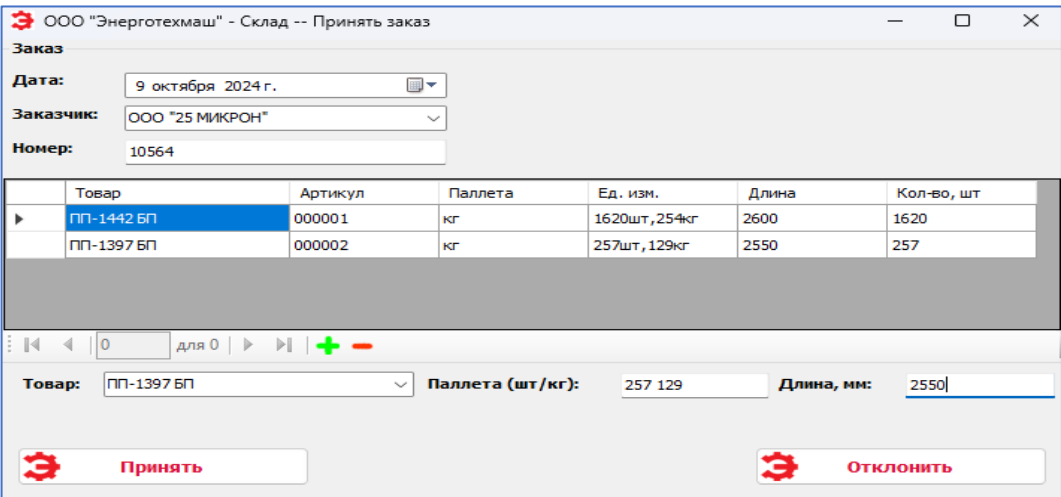


Рисунок 69 – Прием заказов

После нажатия на кнопку «Принять» это окно закроется, а в главном окне появится элемент с информацией о принятом заказе, находящемся в стадии выполнения. В модуле «Заказы» можно посмотреть более подробную информацию, со статусом каждой единицы заказа (рисунок 70).

ИД	Номер	Заказчик	Дата	Состояние
1	10564	1	09.10.2024	Отгружен
2	345	3	09.10.2024	Ожидает распоряжения на отгрузку

Товар	Артикул	Паллета	Состояние
ПП-1442 БП	000001	1620шт, 254кг	Отгружен
ПП-1397 БП	000002	257шт, 129кг	Отгружен

Рисунок 70 – Статус заказа

Теперь менеджер может сформировать распоряжение на отгрузку в модуле «Отгрузка» (рисунок 71).

Заказ	Заказчик	Дата	Товар	Паллета
345	ООО "Алвид"	09.10.2024	ПП-1397 БП	257шт, 129кг

Товар	Заказ	Паллета	Ед. изм.	Длина	Кол. штук
-------	-------	---------	----------	-------	-----------

Распоряжение на отгрузку № от

Заказчик:

Разрешил:

Главбух:

Рисунок 71 – Модуль «Отгрузка»

В верхней таблице находится все выбранные единицы из разных заказов.

Распоряжение, может быть, сформировано из единиц разных заказов при условии, что получатель тот же.

Также можно разбить один заказ на несколько отгрузок. Формирование распоряжения заключается в перетаскивании (с помощью механизма Drag and Drop) единиц заказов в нижнюю таблицу.

После нажатия на кнопку «Создать» названия кнопок изменяться и распоряжение можно распечатать (рисунок 72).

Товар	Заказ	Паллета	Ед. изм.	Длина	Кол. штук
ПП-1442 БП	10564	1620шт, 254кг	кг	2600	1620
ПП-1397 БП	10564	257шт, 129кг	кг	2550	257

Рисунок 72 – Печать распоряжения на отгрузку

При необходимости, распоряжение может быть повторно распечатано в модуле «Распоряжения», в котором отображается список всех распоряжений.

После создания распоряжения на отгрузку, оно появится в главном окне кладовщика в списке справа. При двойном щелчке на распоряжении откроется форма для оформления расходного ордера.

Все поля формы будут заполнены, зоны, откуда брать товар, также будут заполнены программой автоматически, конечно, если указанные товары есть на складе (должны совпадать номенклатурный номер, паллета и длина), в

противном случае ордер не удастся оформить. Аналогично созданию других документов, ордер можно распечатать после создания, или распечатать позже в модуле «Расходные ордера». Распечатанный ордер показан на рисунке 73.

№ п/п	Зона	Артикул	Товар	Заказ	Паллета	Ед. изм.	Длина, мм	Кол-во, шт
1	A	000001	ПП-1442 БП	10564	1620шт,254кг	кг	2600	1620
2	Б	000002	ПП-1397 БП	10564	257шт,129кг	кг	2550	257

Всего 2 наименований

Отпустил _____ (подпись) _____ (должность) **Акимова Марина Владимировна** (Ф.И.О.)

Получил _____ (подпись) _____ (должность) _____ (Ф.И.О.)

Рисунок 73 – Печатная форма расходного ордера

После создания расходного ордера можно наблюдать, что товары из ордера списались со склада (в модуле «Склад» их больше нет). Также можно видеть, что в главном окне «Менеджера» заказ пропал из списка заказов на выполнении, а в модуле «Заказы» выполненный заказ изменил статус на «Отгружен».

Таким образом, процесс тестирования подтвердил, что программное решение функционирует без сбоев, а также выявил случаи, когда система выдает сообщение об ошибочном выполнении определенных команд при обработке запросов.

3.4 Расчет показателей экономической эффективности проекта

Для оценки экономической эффективности разработки программного обеспечения необходимо провести расчет, схожий с составлением небольшого бизнес-плана.

Главный метод расчета заключается в учете затрат, в частности себестоимости разработки [11].

Себестоимость включает несколько основных факторов:

- стоимость ПО (операционная система, среда разработки, СУБД и другие программные средства);
- электроэнергию;
- амортизационные отчисления на оборудование;
- расходы на заработную плату.

Общая сумма затрат формирует себестоимость программного обеспечения, которая должна быть оправданной для организации и учтена в соответствующих статьях расходов.

Затраты на компьютерную конфигурацию представлены в таблице 5.

Таблица 5 – Затраты на компьютерную конфигурацию

Категория затрат	Количество единиц	Стоимость (рублей)
Персональный компьютер	1	30000
Операционная система Windows 10	1	6100
Microsoft SQL Server 2019	1	55790
Visual C++	1	1990
Итого (S_p):		93880

Затраты на создание программного обеспечения включают расходы на электроэнергию, которые определяются на основе времени, затраченного на разработку, мощности оборудования и стоимости 1 кВт электроэнергии, равной 5,04 рубля:

$$S_3 = 0,5 * 80 * 5,04 = 201,60$$

Затраты на амортизацию в течении периода разработки рассчитаем по формуле 1:

$$A_q = \Phi_{\text{перв}} * \frac{A}{F_d} * t, \quad (1)$$

где:

$\Phi_{\text{перв}}$ – первоначальная стоимость компьютерной конфигурации,

A – норма амортизации,

F_d – годовой фонд времени,

t – время, затраченное на разработку.

$$A_q = 93880 * \frac{0,2}{1979} * 80 = 759,01$$

Сложим все затраты и получим, что общие издержки ($S_{\text{общ}}$) составляют:

$$(S_{\text{общ}}) = S_3 + A_q + S_p = 201,60 + 759,01 + 93880 = 94840,61 \text{ руб.}$$

Определим среднюю стоимость труда за час, исходя из месячной зарплаты:

$$C_{\text{ср(час)}} = \frac{C_{\text{ср(мес)}}}{168} = \frac{50000}{168} = 298 \text{ руб./час}$$

В таблице 6 представлена краткая смета на разработку системы автоматизации, демонстрирующая распределение часов и расходов, с учетом отчисления ЕСН с дохода.

Таблица 6 – Смета на разработку системы автоматизации складского учета

Содержание	Количество часов	Стоимость (рублей)
Анализ предметной области	1	298
Постановка задачи	1	298
Разработка технического задания	2	596
Проектирование базы данных	2	596
Разработка интерфейса программы	18	5364
Разработка модулей программы	48	14304
Тестирование системы	8	2384
Итого:	80	23840
ЕСН (рублей)		7152

Итоговая сумма затрат на создание программного обеспечения будет равна общей сумме всех расходов:

$$94840,61 + 23840 + 7152 = 125832,61 \text{ рубля}$$

На основе данных расчетов, общая сумма расходов на создание проекта составила 125832,61 рубля, что наглядно представлено на рисунке 74.

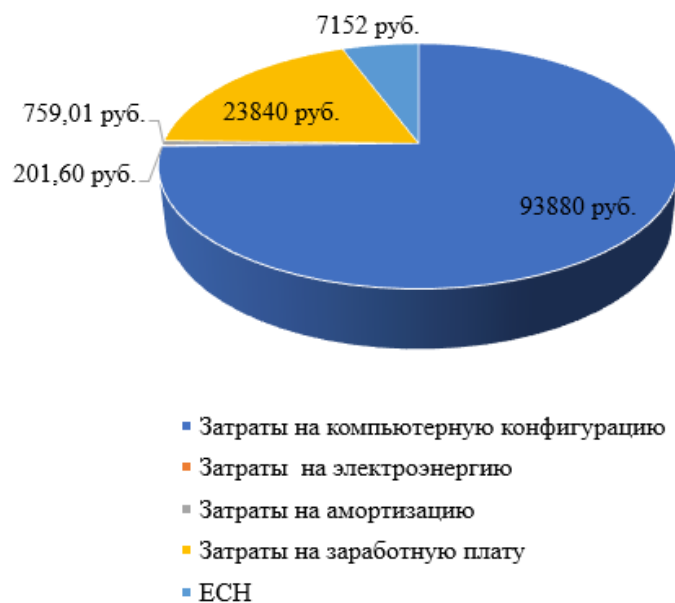


Рисунок 74 – Общая сумма расходов на разработку

Этот анализ позволяет сравнить экономическую эффективность нового приложения с затратами на ручной труд, что важно для определения оправданности инвестиций в проект.

Например, для того, чтобы составить отчет по поступлению и расходу товаров на складе за месяц может в среднем уходить до одного дня рабочего времени. При использовании программного продукта данный временной показатель будет составлять не более двух минут рабочего времени. При этом менеджер или кладовщик склада может получить полный отчет о поступивших и отгруженных товарах.

Приведем пример полученной экономической выгоды от внедрения проекта в расчете на один трудовой год. Экономический эффект с использованием программы представлено в таблице 7.

Таблица 7 – Экономический эффект от автоматизации складского учета

Показатель	Количество операций (штук)	Трудоемкость		Ручной учет операций (мин.)	Учет операций с внедрением системы автоматизации (мин.)
		ручной учет операций (мин.)	учет операций с внедрением системы автоматизации (мин.)		
Учет приходных накладных	1800	10	3	18000	5400
Учет расходных накладных	2000	10	3	20000	6000
Ведение карточки учета товаров	50	5	2	250	100
Составление ведомости учета товаров на складе	12	480	2	5760	24
Итого	3862	505	10	44010	11524

Учитывая стоимость одной минуты рабочего времени, исходя из расчета средней стоимости труда за час, равна 4,97 рубля.

Фонд оплаты труда, который будет потрачен за год при ручном учете операций, будет равен 218729,70 рубля.

Фонд оплаты труда, который будет потрачен за год на учет операций с использованием приложения, будет равен 57274,28 рубля.

Чистая экономия составляет $218729,70 - 57274,28 = 161455,42$ рублей.

Сравнение фонда оплаты труда при ручном и автоматизированном учете операций представлено на рисунке 75.

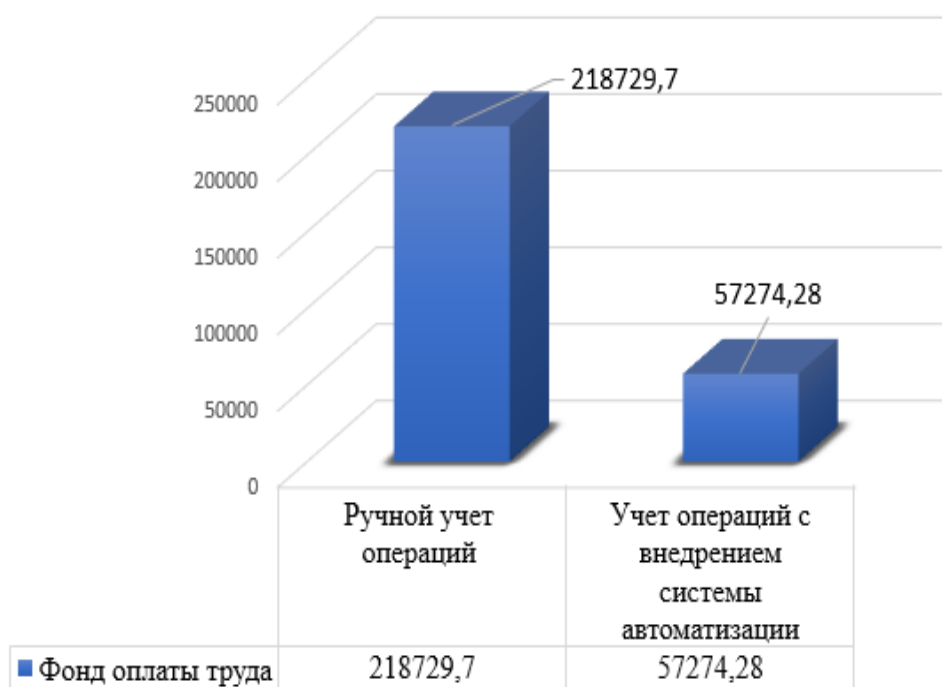


Рисунок 75 – Сравнение фонда оплаты труда при ручном и автоматизированном учете операций с использованием приложения

Таким образом, достигается существенная производительность труда, при которой сохраняется точность и правильность расчета, чего нельзя достичь с помощью расчетов вручную.

Для оценки экономической эффективности проекта рассчитаем срок окупаемости:

$$T_o = \frac{125832,61}{161455,42} * 12 \approx 9 \text{ мес.}$$

Таким образом, на основании полученных расчетов показателей экономической эффективности было показано, что внедрение разрабатываемой системы в бизнес-процессы складского учета является оправданным с экономической точки зрения.

Выводы по главе 3.

В данной главе было проведено физическое проектирование информационной системы (ИС).

Для разрабатываемой ИС выбрана система с «толстым» клиентом. В качестве языка разработки ИС выбран язык программирования C#, имеющий в своем составе такой мощный инструмент, как Entity Framework, который значительно упрощает работу с базой данных. В качестве СУБД выбран MS SQL. Microsoft SQL Server выбран из-за его совместимости с C# и MS Visual Studio.

Описана функциональность и удобство пользовательского интерфейса, что подтверждено успешным проведением функционального тестирования.

Проведенный расчет экономической эффективности показал, что внедрение данной системы в процессы складского учета принесет значительные выгоды.

Заключение

Бакалаврская работа посвящена разработке информационной системе учета товаров на складе.

В выполненной работе рассмотрен полный цикл разработки информационной системы складского учета для склада готовой продукции (СПП) ООО «Энерготехмаш».

Были рассмотрены существующие программные продукты, предназначенные для введения складского учета, и сделан вывод, что они слишком сложны и избыточны для небольшого склада и их адаптация с существующим бизнес-процессом требует больших усилий. Поэтому было принято решение разработать новую систему, полностью ориентированную под нужды именно этого предприятия.

Перед началом проектирования системы было выполнено концептуальное моделирование и разработаны требования к разрабатываемой системе.

На этапе логического проектирования была разработана логическая модель информационной системы, построены диаграмма вариантов использования системы и диаграмма классов, описывающая сущности системы и взаимоотношения между ними. Далее диаграмма классов была преобразована в ER-диаграмму базы данных.

На этапе физического проектирования выбрана архитектура клиент-сервер с использованием «толстого» клиента.

В качестве программного обеспечения информационной системы были выбраны следующие технологии: язык программирования C# в сочетании с фреймворками Windows Forms для создания графического интерфейса пользователя и Entity Framework для работы с данными в базе данных.

В качестве системы управления базой данных был выбран продукт Microsoft SQL Server Express.

Основываясь на разработанной ER-диаграмме, была создана база данных со всеми таблицами и связями, а также сгенерирован SQL-скрипт, позволяющий развернуть базу данных на сервере.

Следующим этапом разработки стало создание проекта приложения в среде разработки MS Visual Studio. Были задействованы возможности Entity Framework по созданию классов предметной области на основе таблиц базы данных, что значительно упростило и ускорило разработку. Затем были созданы экранные формы для взаимодействия пользователей с системой и реализована бизнес-логика приложения.

На финальной стадии разработки было выполнено тестирование созданной системы, показавшее, что полученный программный продукт полностью отвечает поставленным требованиям.

Согласно произведенным расчетам показателей экономической эффективности было показано, что внедрение разрабатываемой системы в бизнес-процессы складского учета является оправданным с экономической точки зрения.

Выполненная работа имеет высокую актуальность в виду бурного развития технологий автоматизации и концепций цифровой экономики, проникновения их во все сферы человеческой деятельности.

Список используемой литературы

1. Анализ и проектирование систем с использованием UML [Электронный ресурс]: URL: <https://urait.ru/bcode/544559> (дата обращения: 15.09.2024).
2. Анисимов В. В. Проектирование информационных систем. [Электронный ресурс]: <https://sites.google.com/site/anisimovkhv/learning/pris>.
3. Бондарь А. Г. Microsoft SQL Server 2022. СПб.: БХВ-Петербург, 2024. 528 с.
4. Вейцман В. М. Проектирование информационных систем: учебное пособие для вузов. М.: Лань, 2022. 216 с.
5. Волк В. К. Базы данных: проектирование, программирование, управление и администрирование: учебник для вузов: для студентов IT специальностей. СПб.: ЛАНЬ, 2023. 244 с.
6. Гагарина Л. Г. Основы проектирования и разработки информационных систем: учебное пособие. М.: ИНФРА-М, 2024. 211 с.
7. Гордеев С. И. Организация баз данных в 2 ч.: учебник. М.: Издательство Юрайт, 2022. 513 с.
8. Горошкина У. Е. Сравнительный анализ подходов к моделированию банковских бизнес-процессов. М.: Федеральное государственное бюджетное образовательное учреждение высшего образования «Российский экономический университет имени Г. В. Плеханова», 2024. 246-253 с.
9. ГОСТ Р 59795-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов. М.: ФГБУ «РСТ», 2021. 32 с.
10. Гусева К. Документирование процессов хранения и отгрузки готовой продукции. Е.: Издательский дом "Ажур", 2022. 121-126 с.
11. Демидов А. В. Оценка экономической эффективности ИТ-проектов: Учебно-методическое пособие. М.: Прометей, 2023. 122 с.

12. Документирование нефункциональных требований с использованием FURPS+ [Электронный ресурс]. URL: https://www.marcinziemek.com/blog/content/articles/8/article_en.html (дата обращения: 06.10.2024).
13. Коваленко В. В. Проектирование информационных систем: учебное пособие 2-е изд. М.: ИНФРА-М, 2023. 357 с.
14. Комаров В. И. Путеводитель по базам данных. М.: ДМК-Пресс, 2024. 520 с.
15. Кузнецов С. Д. Базы данных. Модели и языки. М.: Бином-Пресс, 2021. 720 с.
16. Методология функционального моделирования IDEF0. Руководящий документ. Официальное изд. М.: ИПК Издательство стандартов, 2000. 76 с.
17. Михеева Е. В. Информационные технологии в профессиональной деятельности: учебное издание. М: Академия, 2023. 416 с.
18. Осипов Д. Л. Технологии проектирования баз данных. М.: ДМК Пресс, 2019. 498 с.
19. Проектирование информационных систем [Электронный ресурс]: URL: <https://urait.ru/bcode/489307> (дата обращения: 17.09.2024).
20. Проектирование информационных систем [Электронный ресурс]: URL: <https://urait.ru/bcode/556554> (дата обращения: 17.09.2024).
21. Проектирование информационных систем [Электронный ресурс]: URL: <https://znanium.com/catalog/product/1894610> (дата обращения: 15.09.2024).
22. Разработка программного обеспечения [Электронный ресурс]: URL: https://ru.wikipedia.org/wiki/Разработка_программного_обеспечения (дата обращения: 25.09.2024).
23. Стельмашонок Е. В. Моделирование процессов и систем: учебник и практикум для вузов. М.: Издательство Юрайт, 2024. 304 с.

24. Тестирование белого ящика и черного ящика [Электронный ресурс]: URL: <https://unetway.com/tutorial/testirovanie-belogo-asika-i-chernogo-asika> (дата обращения: 25.09.2024).
25. Тестирование программного продукта [Электронный ресурс]: URL: https://www.tadviser.ru/index.php/Статья:Тестирование_программного_продукта (дата обращения: 25.09.2024).
26. Чистов Д. В. Проектирование информационных систем: Учебник и практикум для вузов. М.: Юрайт, 2022. 258 с.
27. Шитов В. И. Основы проектирования баз данных: учебное пособие. М.: ИНФРА- М, 2023. 236 с.
28. Date C. J., A. Kannan and S. Swamynathan, An Introduction to Database Systems, Pearson Education, Eight Edition, 2009. 1024 p.
29. Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel, UML @ Classroom: An Introduction to Object-Oriented Modeling (Undergraduate Topics in Computer Science), Springer; 2015. Edition, 2015. 220 p.
30. Michael Kirshteyn, Mastering 3NF. A Comprehensive Guide to Efficient Relational Database Design (Data Modeling.), 2024, 150 p.
31. Russ Miles, Kim Hamilton, Learning UML 2.0: A Pragmatic Introduction to UML, O'Reilly and Associates, 2006. 286 p.
32. Sikha Bagui, Richard Earp, Database Design Using Entity-Relationship Diagrams (Foundations of Database Design), Auerbach Publications, 2011. 371 p.