

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

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

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Бизнес-информатика

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка информационной системы управления складскими запасами на предприятии ООО СДТ»

Обучающийся О.В. Огольцова _____
(И.О. Фамилия) (личная подпись)

Руководитель А.П. Тонких _____
(ученая степень, звание, И.О. Фамилия)

Тольятти 2022

Аннотация

Выпускная квалификационная работа 69 страниц, 44 рисунка, 7 таблиц, 29 источников литературы.

СКЛАД, ЗАПАСЫ, СКЛАДСКИЕ ЗАПАСЫ, ТОВАРНО-МАТЕРИАЛЬНЫЕ ЦЕННОСТИ, ИНФОРМАЦИОННАЯ СИСТЕМА

Темой выпускной квалификационной работы является разработка информационной системы управления складскими запасами для отдела материально-технического обеспечения, предназначенная для повышения эффективности работы сотрудников, занятых ведением учета материальных запасов.

Отдел МТО непрерывно пользуется информацией о поставщиках, товарах, поставках и т.д. До сих пор остались предприятия, которые используют бумажные журналы учета для ведения материально-технического обеспечения. А это, как правило, приводит к ошибкам и задержкам в обработке данных, которые в итоге могут принести убыток компании. Поэтому тема настоящей работы является актуальной.

Цель работы – разработка информационной системы управления складскими запасами на предприятии.

Задачи работы:

- исследовать предметную область и выполнить постановку задачи на разработку информационной системы управления складскими запасами;
- спроектировать информационную систему, разработать структуру базы данных;
- разработать информационную систему, руководство пользователя.

В первом разделе работы исследована деятельность сотрудников отдела материально-технического обеспечения по управлению складскими запасами.

Рассмотрены основные функции и задачи отдела материально-технического обеспечения. Выполнено моделирование предметной области с помощью методологии IDEF0.

Второй и третий раздел включают описание информационной модели. Представлен состав и структура классификаторов. Рассмотрены нормативно-справочные документы, реквизитный состав входных и выходных документов. Описана структура базы данных.

В третьем разделе описан процесс разработки информационной системы, рассмотрен визуальный интерфейс информационной системы, представлена схема технологического процесса сбора информации.

Содержание

Введение.....	5
1 Исследование процессов управления складскими запасами на предприятии7	
1.1 Анализ деятельности отдела материально-технического обеспечения... 7	
1.2 Выделение бизнес-процессов организации	9
1.3 Постановка задачи.....	15
1.4 Анализ существующих разработок	17
2 Проектирование информационной системы управления складскими запасами на предприятии	22
2.1 Выбор технического обеспечения и средств разработки.....	22
2.2 Характеристика входной и выходной информации	24
2.3 Описание структуры базы данных	26
3 Разработка информационной системы управления складскими запасами на предприятии.....	32
3.1 Дерево функций и описание интерфейса информационной системы управления запасами на предприятии	32
3.2 Структурная схема разрабатываемой системы	35
3.3 Описание программных модулей	37
3.4 Схема работы информационной системы управления складскими запасами на предприятии	42
3.5 Руководство пользователя	44
Заключение	52
Список используемой литературы	54
Приложение А Фрагменты исходного кода	57

Введение

Данная работа посвящена разработке информационной системы (ИС) управления складскими запасами на предприятии.

Учетом складских (материальных) запасов, как правило, занимается отдел материально-технического обеспечения (МТО).

Отдел материально-технического обеспечения является стратегически важным отделом для любой организации. Без своевременной закупки расходных материалов может быть прекращена работа всех отделов. Если же предприятие является производственным, то без качественной работы отдела материально-технического обеспечения может быть прерван непрерывный производственный процесс.

Отдел МТО непрерывно пользуется информацией о поставщиках, товарах, поставках и т.д. До сих пор остались предприятия, которые используют бумажные журналы учета для ведения материально-технического обеспечения. А это, как правило, приводит к ошибкам и задержкам в обработке данных, которые в итоге могут принести убыток компании. Поэтому тема настоящей работы является актуальной.

Цель работы – разработка информационной системы управления складскими запасами на предприятии.

Задачи работы:

- исследовать предметную область и выполнить постановку задачи на разработку информационной системы управления складскими запасами;
- спроектировать информационную систему, разработать структуру базы данных;
- разработать информационную систему, руководство пользователя.

Объектом исследования является компания, которая ведет учет складских запасов.

Предметом исследования является разработка информационной системы управления складскими запасами.

Выпускная квалификационная работа состоит из введения, трех разделов, заключения, списка используемых источников.

Первый раздел включает описание постановки задачи на разработку информационной системы, а также анализ деятельности автоматизируемого предприятия. Обоснована необходимость автоматизации исследуемого процесса.

Во втором разделе выполнено проектирование информационной системы, разработана структура базы данных (БД).

В третьем разделе работы приведено описание разработки информационной системы. Разработано техническое, программное и информационное обеспечение информационной системы. Приведено руководство пользователя.

Практическая значимость определяется возможностью внедрения разработанного программного обеспечения в любой организации, в которой ведется учет складских запасов.

1 Исследование процессов управления складскими запасами на предприятии

1.1 Анализ деятельности отдела материально-технического обеспечения

Предметной областью настоящей выпускной квалификационной работы является деятельность отдела материально-технического обеспечения по управлению складскими запасами.

Под учетом складских запасов понимается совокупность операций по закупке, доставке, приемке и хранению товарно-материальных ценностей.

На рисунке 1 представлены основные функции отдела материально-технического обеспечения (МТО) [4].

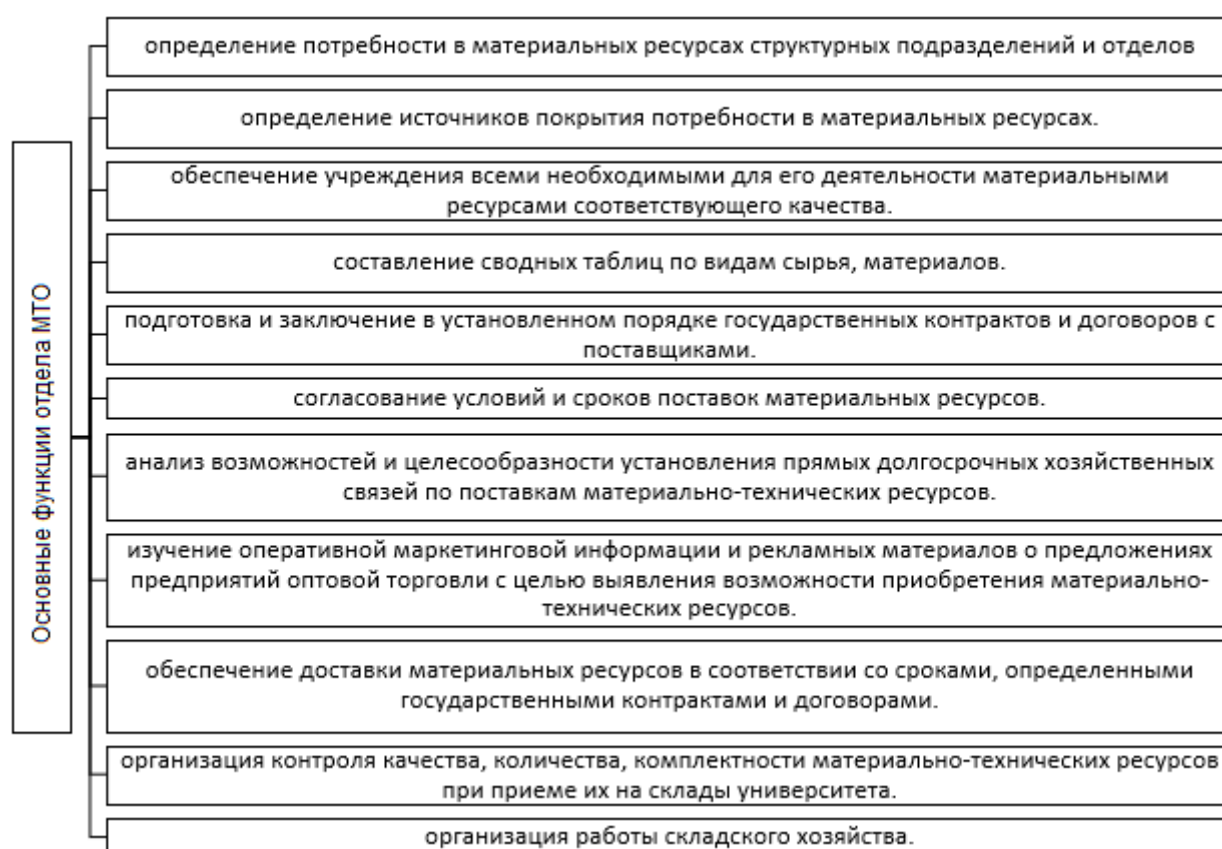


Рисунок 1 – Основные функции отдела МТО

На рисунке 2 представлены функциональные обязанности специалистов по снабжению [4].

Специалист по снабжению руководствуется в своей деятельности [3]:

- Уставом компании, который регламентирует особые варианты расчеты с поставщиками и другие условия сотрудничества;
- приказами, постановлениями и законами Российской Федерации, регулирующими процессы снабжения компании необходимыми товарно-материальными ценностями;
- приказами и распоряжениями руководства в части снабжения;
- должностной инструкцией специалиста по снабжению, которая регламентирует порядок работы специалиста.

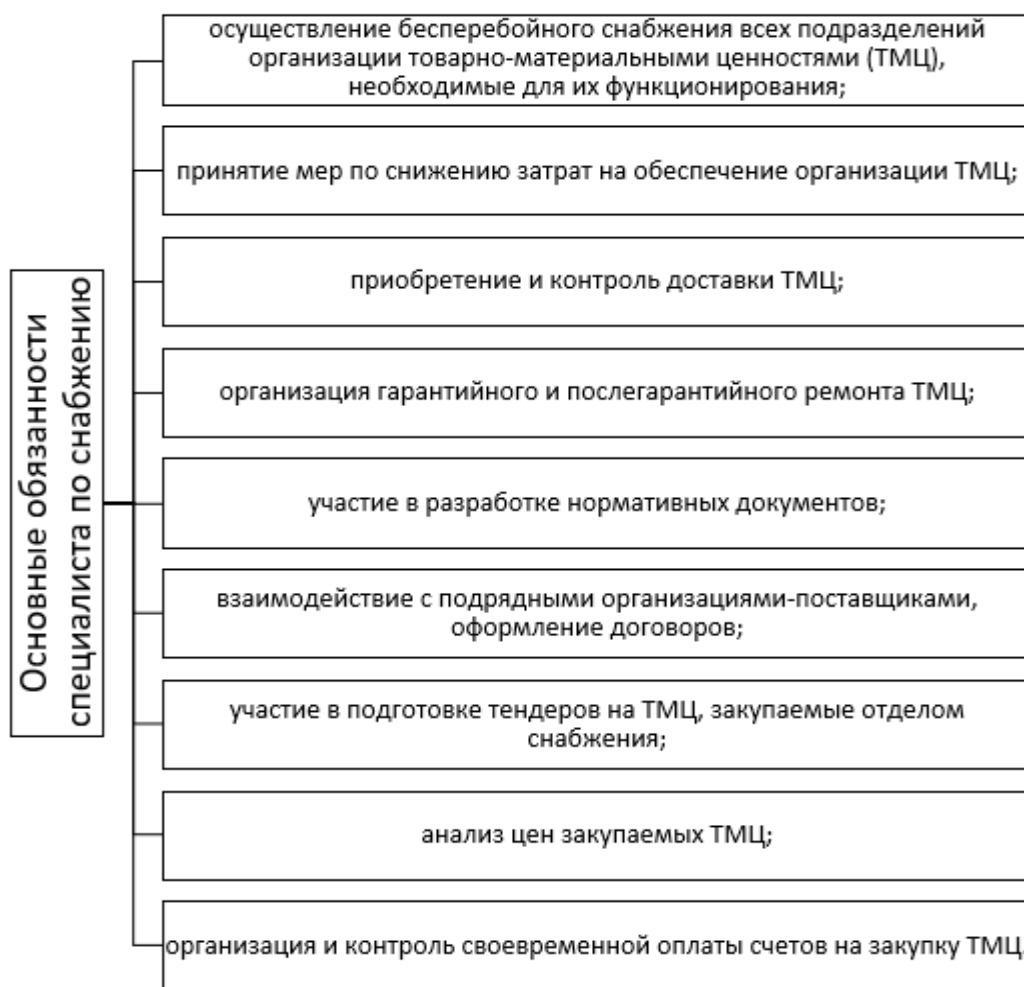


Рисунок 2 – Основные обязанности специалиста по снабжению

Компания заключает договоры с поставщиками, которые в назначенный срок поставляют товарно-материальные ценности на склады организации для обеспечения непрерывного производства и поддержания деятельности организации.

1.2 Выделение бизнес-процессов организации

Рассмотрим текущие бизнес-процессы деятельности специалистов по снабжению по учету материальных запасов с помощью методологии IDEF0 [10, 20]. Контекстная диаграмма представлена на рисунке 3.

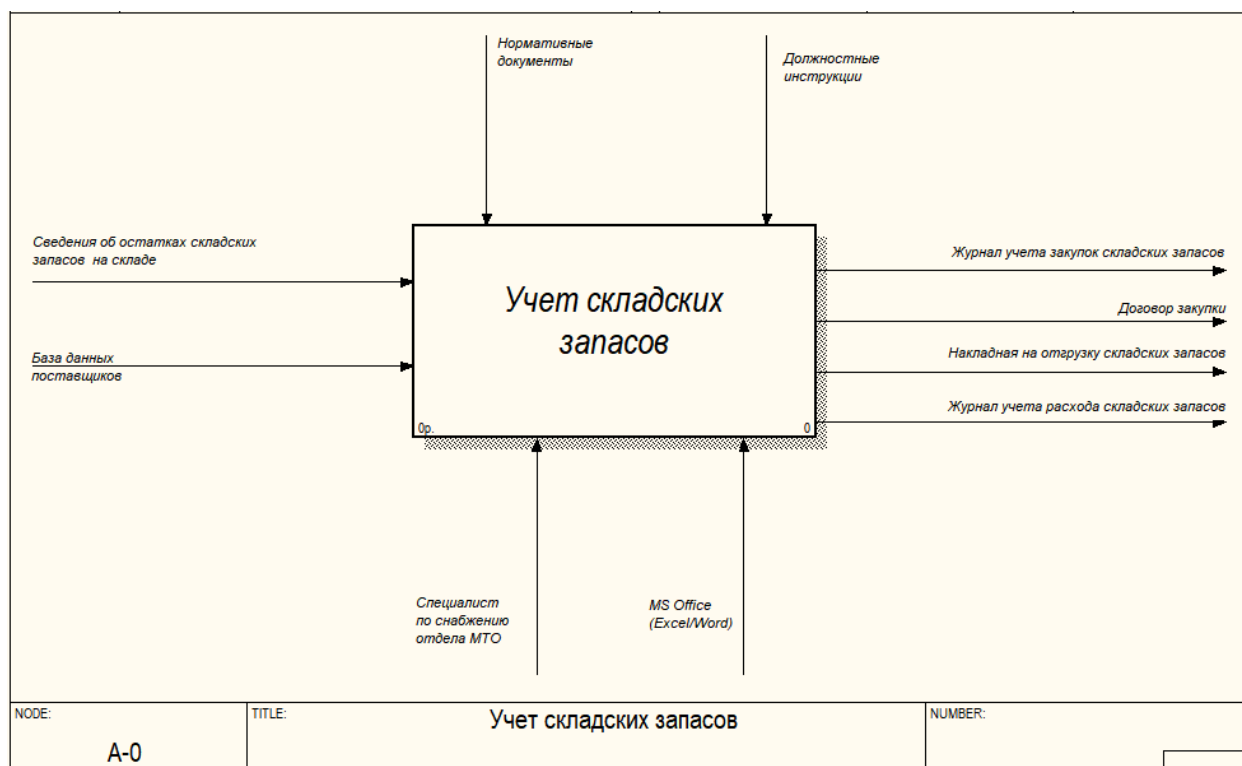


Рисунок 3 – Контекстная диаграмма «как есть»

К складским запасам относятся химические реагенты для производства, канцелярские товары, необходимые для обеспечения деятельности

организации, и любые другие товарно-материальные ценности, которые необходимы для осуществления деятельности организации.

На входе в исследуемый процесс представлена база данных поставщиков, которая используется при закупке необходимых товарно-материальных ценностей.

Выходными потоками являются журнал учета закупок, договор закупки, заявка на закупку, накладная на отгрузку складских запасов, журнал учета расхода складских запасов.

В процессе учета складских запасов принимают участие специалисты по снабжению отдела МТО. Для ведения учета используются офисные программы MS Office (Word/Excel).

Проведя декомпозицию, была получена следующая IDEF0-модель, представленная на рисунке 4.

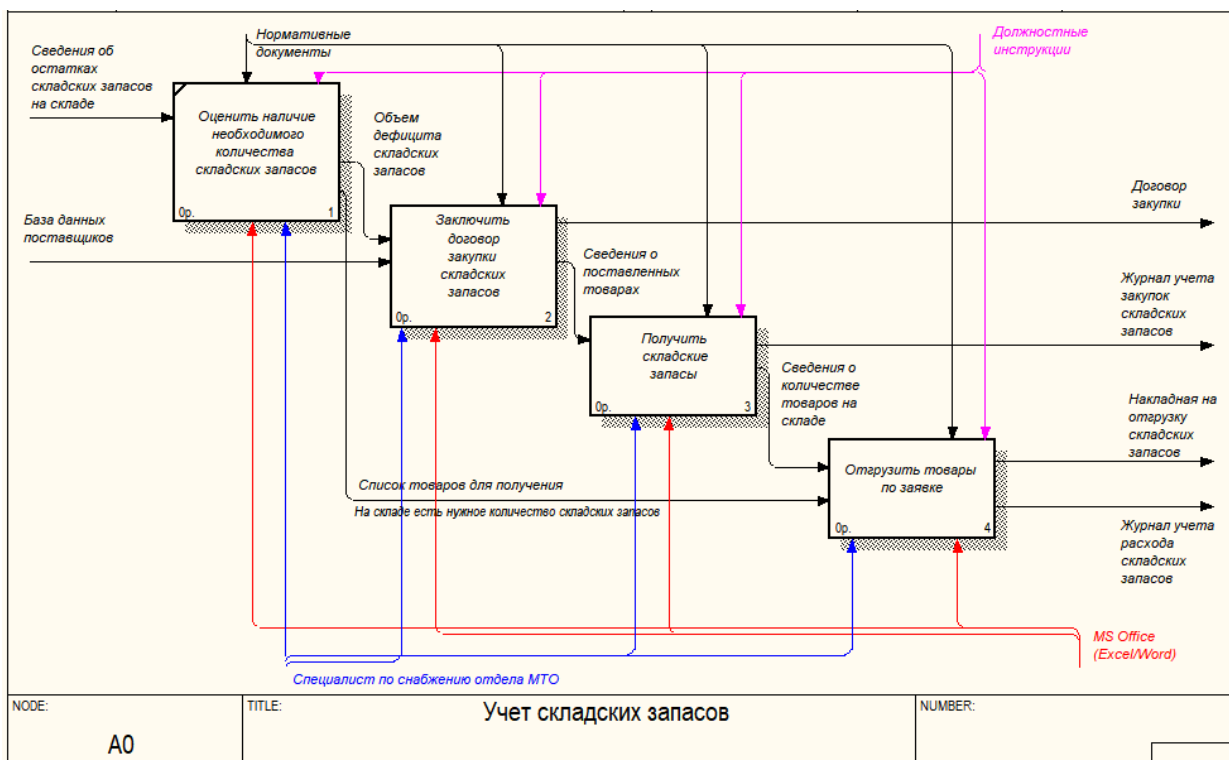


Рисунок 4 – Декомпозиция модели «как есть»

Основные бизнес-процессы ведения учета складских запасов:

- оценить наличие необходимого количества складских запасов;
- заключить договор закупки складских запасов;
- получить товары на склад;
- провести учет расхода складских запасов;
- сформировать отчетность.

Заявки на отгрузку складских заявок поступают специалистам отдела материально-технического обеспечения, которые в свою очередь собирают товар к отгрузке и выдают товар со склада, оформляют отчетность о расходе складских запасов.

Описание процесса заключения договора представлено на рисунке 5.

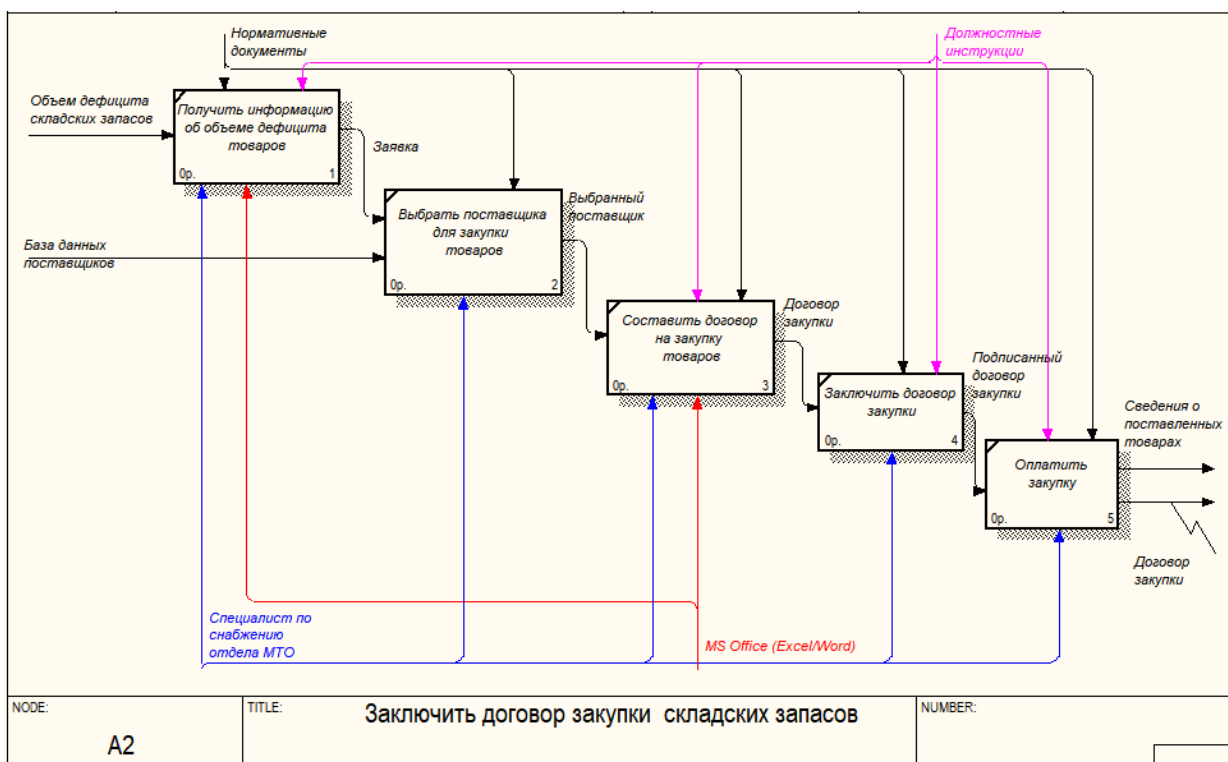


Рисунок 5 – Декомпозиция процесса «ЗаклЮчить договор закупки складских запасов»

При получении заявки на закупку товарных материальных ценностей специалист по снабжению отдела МТО анализирует цены поставщиков на

нужные товары. Если условия устраивают, то с поставщиком согласовывается окончательная стоимость закупки, и в итоге заключается договор поставки.

Оплата по договору поставки выполняется бухгалтером организации.

На рисунке 6 представлена декомпозиция блока «Получить материалы на склад». Если товар принят без расхождений, то он сразу же оприходывается на склад и размещается в складских помещениях. Если же были расхождения в поставке с договором или накладной, то оформляется документ о расхождениях и претензия поставщику.

Любое подразделение может запросить получение товарно-материальных ценностей со склада по заявке, которая утверждается руководством. Обычно требуется несколько подписей вышестоящего руководства и начальника склада.

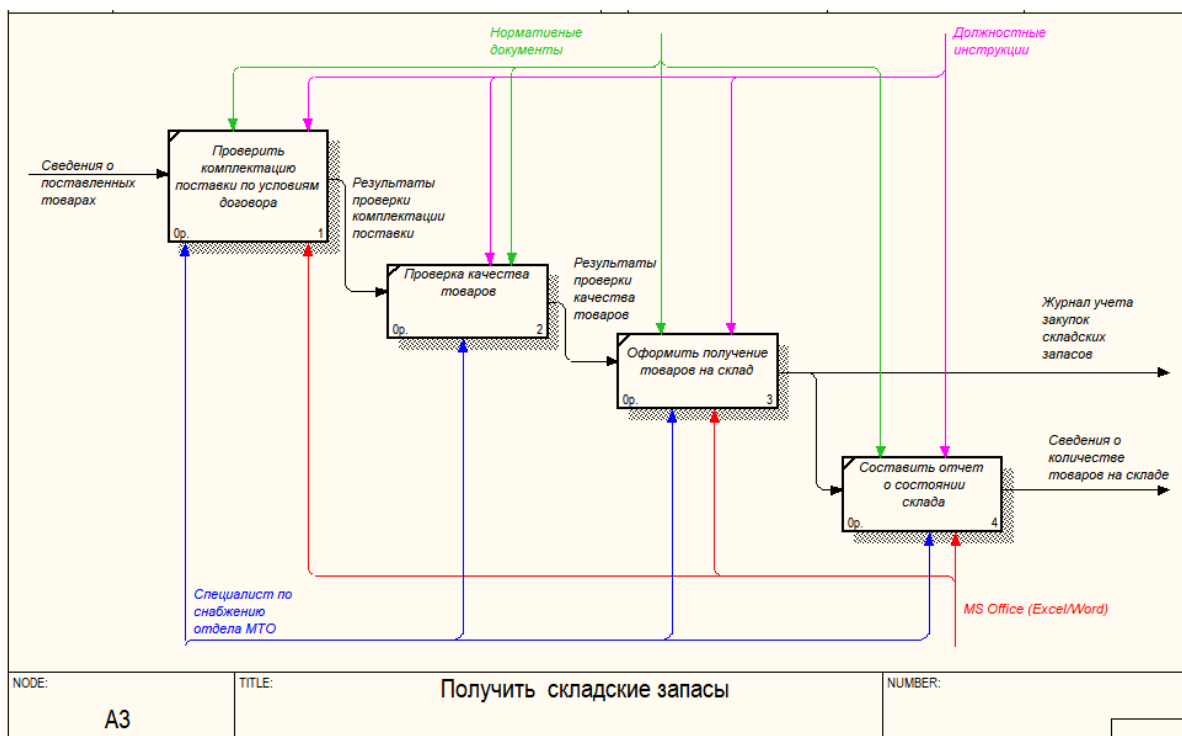


Рисунок 6 – Описание процесса получения складских запасов

Описание процесса отгрузки товаров по заявке представлено на рисунке

7.

Целесообразность разработки информационной системы учета складских запасов определяется следующими выводами:

- разработка и внедрение информационной системы позволит сократить количество ручных операций по ведению журналов учета поставок и расхода складских запасов;
- после внедрения информационной системы формирование отчетных материалов будет выполняться без участия специалистов отдела МТО;
- исключение ручной обработки информации снизит вероятность погрешностей при анализе данных;
- после внедрения модуля информационной системы сократится количество бумажных документов.

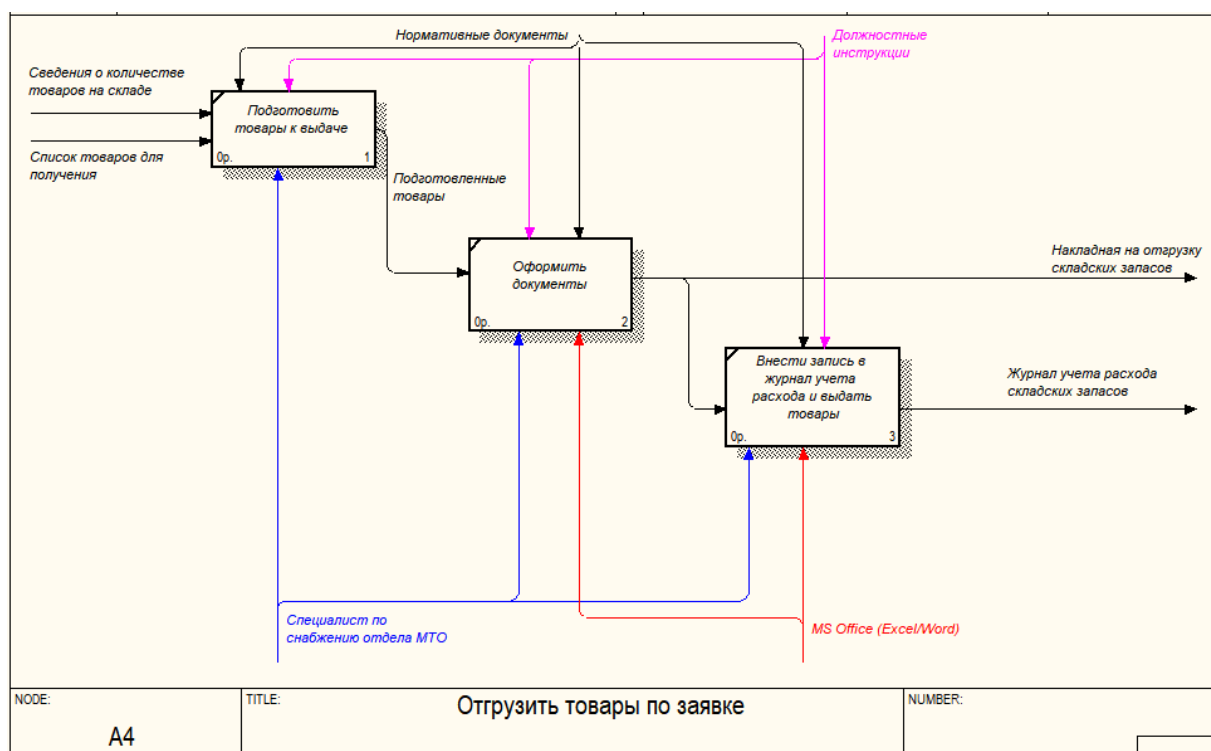


Рисунок 7 – Описание процесса отгрузки товаров по заявке

Внесение информации о поступлении и расходе товаров занимает около 1 часа. После внедрения информационной системы время на ведение учета

складских запасов значительно сократится. Прогнозируется, что производительность работы сотрудников по внесению информации о закупке и расходе складских запасов будет увеличена более чем на 60%.

На рисунке 8 приведена декомпозиция комплекса задач по учету складских запасов.

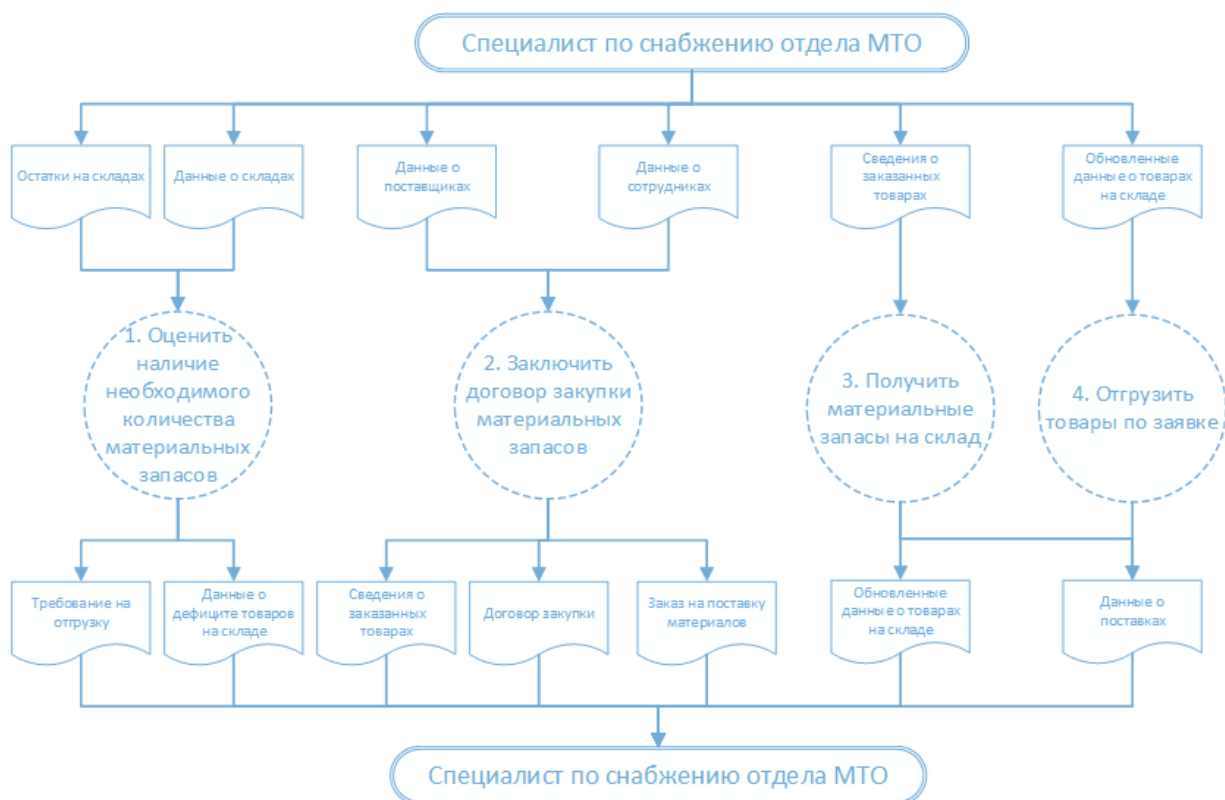


Рисунок 8 – Декомпозиция комплекса задач по учету складских запасов

Входные данные:

- остатки на складах;
- информация о складах;
- данные о сотрудниках;
- данные о поставщиках;
- обновленные данные о товарах на складе.

1.3 Постановка задачи

На основе проведенного исследования процесса учета складских запасов были выявлены следующие проблемы:

- заполнение журналов вручную;
- поиск информации занимает слишком много времени;
- обработка накопленной информации и формирование отчетов требует участия нескольких сотрудников;
- инвентаризация склада выполняется вручную.

Предполагается, что разработка информационной системы учета складских запасов позволит получить следующие результаты:

- сотрудники отдела МТО будут тратить меньше времени на заполнение журналов учета или вовсе избавятся от них;
- при учете данных с помощью модуля ИС будет контролироваться корректность внесенных данных;
- расчет остатков будет выполняться с помощью модуля ИС;
- снизить затраты на анализ данных;
- позволить выполнять расширенный поиск информации;
- защитить информацию.

Для устранения найденных проблем было принято решение о разработке автоматизированной информационной системы управления складскими запасами предприятия.

Целью проекта является разработка информационной системы для автоматизации учета складских запасов. Благодаря внедрению информационной системы значительно повысится уровень качества работы отдела МТО.

Для реализации информационной системы необходимо [10, 12]:

- спроектировать архитектуру и структуру информационной системы;
- разработать алгоритм, описывающий процесс учета оказываемых услуг;

- создать интерфейс информационной системы.

Назначение информационной системы:

- хранить и накапливать информацию о договорах на поставку товаров;
- хранить и накапливать информацию о движении складских запасов;
- хранить и накапливать информацию о поставках складских запасов;
- хранить и накапливать информацию об ответственных сотрудниках;
- формировать отчетность.

С информационной системой будет работать специалист по снабжению отдела МТО, а также администратор модуля ИС, который будет редактировать сведения о пользователях модуля.

На рисунке 9 представлена диаграмма вариантов использования проектируемого модуля [5, 8].

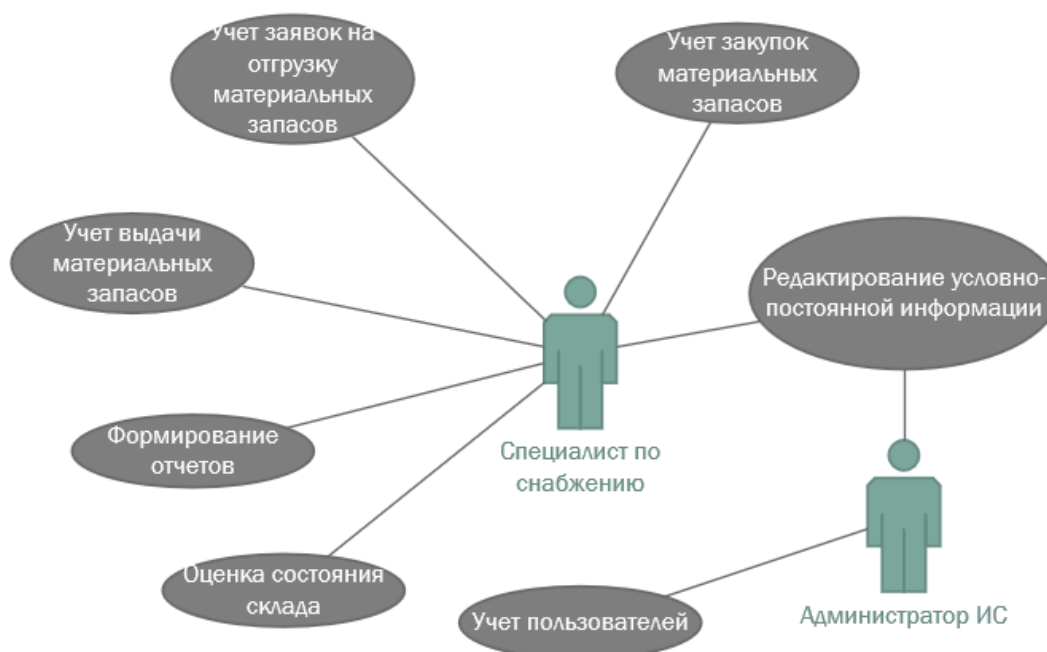


Рисунок 9 – Диаграмма вариантов использования

Основные функции информационной системы:

- учет закупок товаров;

- учет заявок на отгрузку товаров;
- оценка состояния склада;
- учет выдачи товаров;
- формирование отчетной документации.

Предполагается, что модуль ИС будет использоваться ежедневно в течение рабочего времени.

Пользователи информационной системы:

- администратор информационной системы;
- специалист по снабжению.

1.4 Анализ существующих разработок

Самыми популярными системами управления складскими запасами (закупками) являются:

- Онлайн-портал управления закупками AGORA [27];
- АИС «Комита» [28];
- PROF-IT SRM: Управление закупками [29].

AGORA – это набор современных инструментов для автоматизации процесса закупок и взаимодействия с поставщиками на всех этапах сделки.

Пример визуального интерфейса системы AGORA представлен на рисунке 10.

Отличительные особенности системы:

- корпоративный сайт закупок;
- планирование закупок;
- управление документацией;
- управление качеством и рейтингом контрагентов;
- интеграция и настройка ERP, CRM и аналитики;
- персонализированный личный кабинет;

- подбор поставщиков по параметрам в зависимости от типа торговой процедуры;
- проведение торгов с различными этапами согласования и обработки.

The screenshot displays the 'Заказы' (Orders) section of the AGORA system. It features a search bar at the top and a table of order records. The table columns include: Заказ (Order), Покупатель (Buyer), Доставка (Delivery), Статус (Status), Дата отгрузки (Shipping Date), Последний корректировщик (Last Modifier), Товары (Goods), Завод отгрузки (Factory), and Комментарий (Commentary). The orders listed are for 'ЗАО БКК "КОЛОМЕНСКИЙ"' with various quantities and dates ranging from 2019 to 2020. Statuses include 'Доставка' (Delivery), 'Подтверждено' (Confirmed), and 'Отменено' (Cancelled).

Заказ	Покупатель	Доставка	Статус	Дата отгрузки	Последний корректировщик	Товары	Завод отгрузки	Комментарий
№ 16	ЗАО БКК "КОЛОМЕНСКИЙ"	2	Доставка	13 января 2020 г.	Мария Сотникова	RU2P (Узловая)		
№ 15	ЗАО БКК "КОЛОМЕНСКИЙ"	1	Доставка	9 января 2020 г.	Мария Сотникова	RU2P (Узловая)		
№ 14	ЗАО БКК "КОЛОМЕНСКИЙ"	2	Доставка	10 января 2020 г.	Мария Сотникова	RU2P (Узловая)		
№ 13	ЗАО БКК "КОЛОМЕНСКИЙ"	2	Доставка	8 января 2020 г.	Мария Сотникова	RU2P (Узловая)		
№ 12	ЗАО БКК "КОЛОМЕНСКИЙ"	2	Доставка	4 января 2020 г.	Мария Сотникова	RU2P (Узловая)		
№ 11	ЗАО БКК "КОЛОМЕНСКИЙ"	3	Доставка	30 декабря 2019 г.	Мария Сотникова	RU2P (Узловая)		
№ 9	ЗАО БКК "КОЛОМЕНСКИЙ"	1	Доставка	26 декабря 2019 г.	Мария Сотникова	RU2P (Узловая)		
№ 8	ЗАО БКК "КОЛОМЕНСКИЙ"	0	Доставка	30 декабря 2019 г.	Мария Сотникова	RU2P (Узловая)		
№ 7	ЗАО БКК "КОЛОМЕНСКИЙ"	2	Доставка	23 декабря 2019 г.	Мария Сотникова	RU2P (Узловая)		
№ 6	ЗАО БКК "КОЛОМЕНСКИЙ"	2	Доставка	30 декабря 2019 г.	Мария Сотникова	RU2P (Узловая)		
№ 5	ЗАО БКК "КОЛОМЕНСКИЙ"	1	Доставка	24 декабря 2019 г.	Ирина Алимурадова	RU2P (Узловая)		
№ 4	ЗАО БКК "КОЛОМЕНСКИЙ"	2	Доставка	27 декабря 2019 г.	Мария Сотникова	RU2P (Узловая)		
№ 3	ЗАО БКК "КОЛОМЕНСКИЙ"	3	Доставка	25 декабря 2019 г.	Ирина Алимурадова	RU2P (Узловая)		
№ 2	ЗАО БКК "КОЛОМЕНСКИЙ"	2	Доставка	23 декабря 2019 г.	Мария Сотникова	RU2P (Узловая)		
№ 1	ЗАО БКК "КОЛОМЕНСКИЙ"	0	Доставка	10 декабря 2019 г.	Ирина Алимурадова	RU2P (Узловая)		тестовый з...

Рисунок 10 – Пример визуального интерфейса системы AGORA

Автоматизированная информационная система «Комита» предназначена для автоматизации и информационно-аналитической поддержки деятельности по осуществлению закупок: основных операций планирования и формирования, размещения и исполнения государственного и муниципального заказа.

АИС «Комита» разработана на основе технологии Web 2.0, что позволяет работать через стандартные Web-браузеры без установки специального ПО на рабочие места пользователей.

Пример визуального интерфейса системы представлен на рисунке 11.

АИС «Комита» выполняет следующие основные функции:

- планирование закупок (документы План закупок, План-график);
- ведение реестра закупок малого объема;

- функционирование справочников, проверок заполнения электронных документов и контроля правомерности действий заказчиков;
- настройка внутреннего документооборота;
- формирование аналитических отчетов.

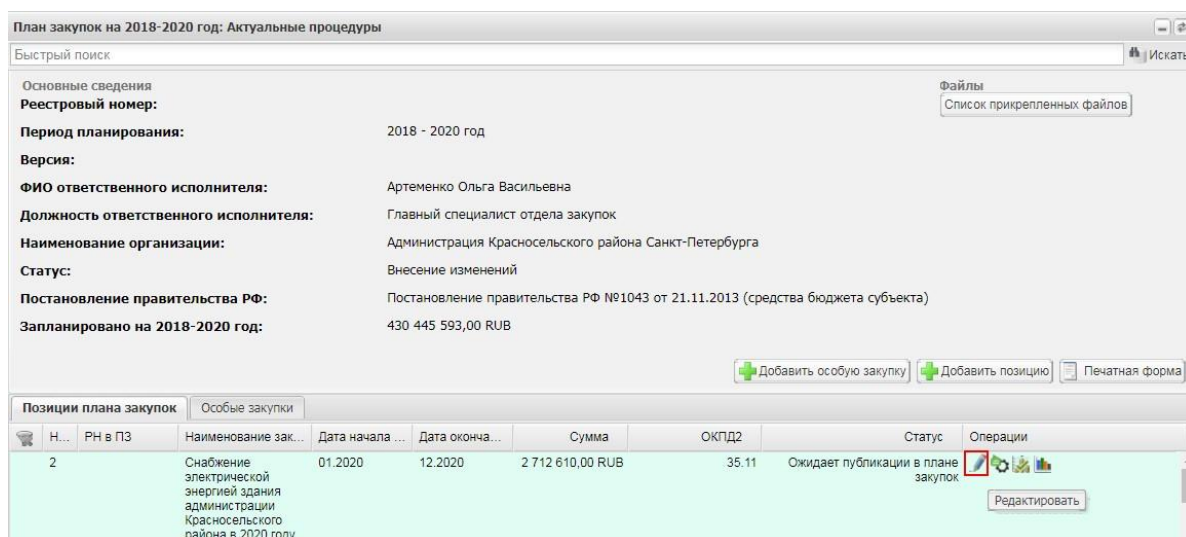


Рисунок 11 – Пример визуального интерфейса АИС «Комита»

«PROF-IT SRM: Управление закупками» предлагает возможности по управлению процессом поставок на всех значимых этапах их реализации: от анализа потребностей компании до подготовки отчетности о работе с поставщиками.

Пример визуального интерфейса системы представлен на рисунке 12.

«PROF-IT SRM: Управление закупками» – это современное решение для управления взаимодействием с поставщиками. SRM-система включает набор полезных инструментов, например, для формирования плана закупок, регистрации поставщиков, визуализации данных по контрольным точкам закупочной деятельности и др. Система обеспечивает автоматизацию управления закупками в четком соответствии с требованиями 223-ФЗ и 44-ФЗ.

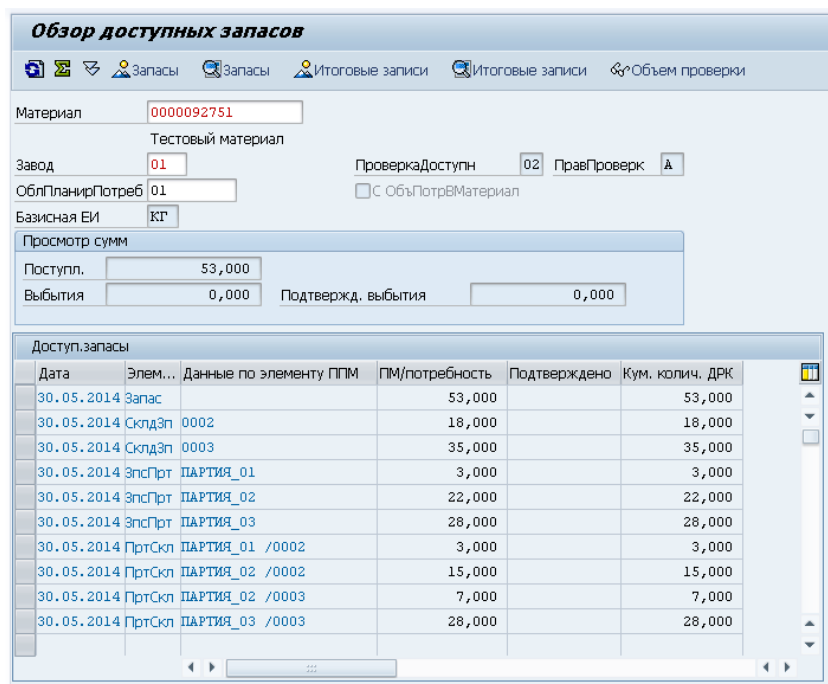


Рисунок 12 – Пример визуального интерфейса системы «PROF-IT SRM: Управление закупками»

Результаты проведенного анализа систем управления закупками представлены в таблице 1.

Таблица 1 – Сравнение систем управления закупками

Показатель сравнения	AGORA	Комита	PROF-IT SRM: Управление закупками
1	2	3	4
Вид интерфейса	Веб-приложение	Веб- и Windows приложение	Windows приложение
Возможность ведения учета закупок товаров	Поддерживается	Поддерживается	Поддерживается
Возможность ведения учета отгрузки товаров со склада	Поддерживается	Поддерживается	Поддерживается
Возможность создания отчетности	Присутствует	Присутствует	Присутствует
Уровень защиты данных	Высокий	Средний	Средний

Продолжение таблицы 1

1	2	3	4
Расширяемость с помощью подключения дополнительных модулей	Присутствует	Присутствует	Отсутствует
Возможность оперативной помощи техническими специалистами	Присутствует	Отсутствует	Отсутствует
Стоимость приобретения	Более 300 000 руб.	Более 450 000 руб.	Более 200 000 руб.

Не все рассмотренные системы соответствуют заявленным требованиям к информационной системе управления складскими запасами, поэтому для решения поставленной задачи информационная система автоматизации управления складскими запасами будет разработана собственными силами.

Выводы по первому разделу

В первом разделе работы проведено исследование деятельности сотрудников отдела материально-технического обеспечения. Проведено моделирование бизнес-процессов, выявлены проблемные моменты, которые могут быть решены внедрением информационной системы. Выполнена постановка задачи на разработку информационной системы. Проведен анализ готовых информационных систем, которые могут быть использованы для решения поставленной задачи и обоснована необходимость разработки новой системы.

2 Проектирование информационной системы управления складскими запасами на предприятии

2.1 Выбор технического обеспечения и средств разработки

При выборе программного обеспечения для разработки информационной системы автор основывался на своем опыте, а также популярности используемых средств.

На рынке программного обеспечения наиболее популярными СУБД являются: PostgreSQL; MySQL; Microsoft SQL Server; Oracle [16].

В таблице 2 представлена сравнительная характеристика стоимости СУБД.

Таблица 2 – Сравнение стоимости СУБД

СУБД	Версия	Стоимость, руб.
PostgreSQL	14.2	98000
MySQL	8.0.29	0
MS SQL Server	Express 2022 Standard 2022	0 180000
Oracle	21c	250000

В качестве СУБД выбрана современная и мощная система MySQL.

В таблице 3 представлена сравнительная характеристика стоимости средств разработки.

Таблица 3 – Сравнительная характеристика стоимости средств разработки

Среда разработки	Номер версии	Основные языки программирования	Стоимость приобретения, руб.
1	2	3	4
MS Visual Studio	2022	C#, F#, Visual C++, XAML, JavaScript	165000

Продолжение таблицы 3

1	2	3	4
NetBeans	12	C, C++, PHP, Java	0
PyCharm	2022.02	Python, Node.js, JavaScript	185000
Code:: Blocks	2022	C, C++	85 000

Для разработки клиентской части информационной системы выбрана среда разработки MS Visual Studio.

В качестве языка программирования будет использован язык C#.

Диаграмма развертывания [16, 23] информационной системы учета складских запасов представлена на рисунке 13.

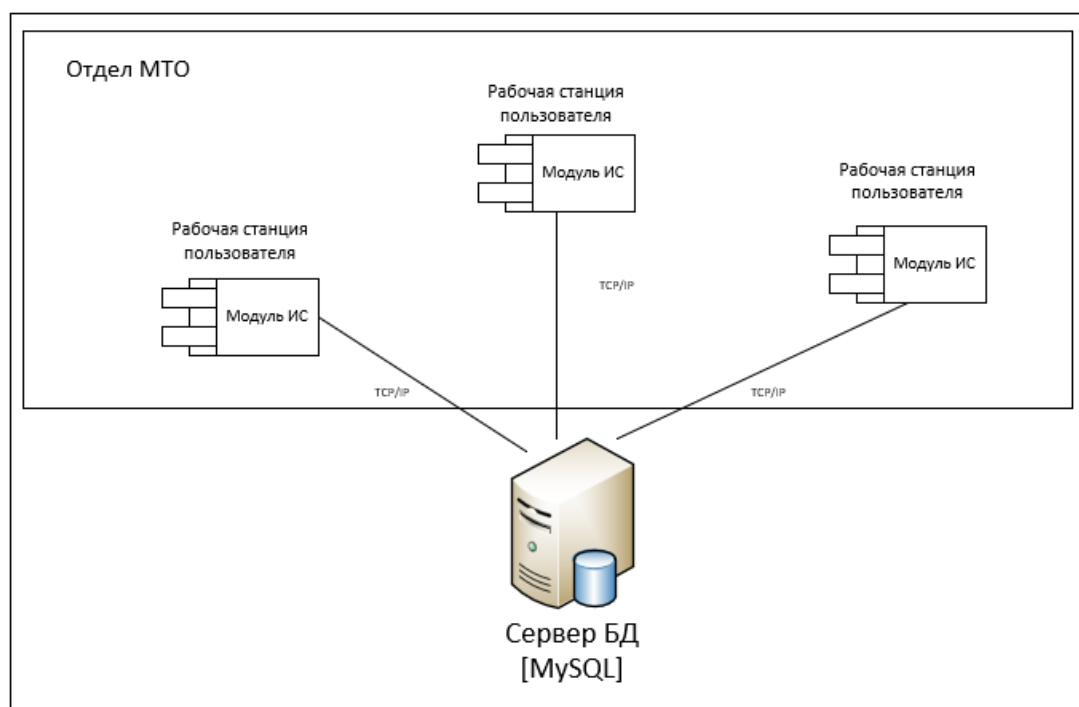


Рисунок 13 – Диаграмма развертывания

Минимальные технические требования персональных компьютеров, на которых будет установлена клиентская часть информационной системы:

- оперативная память не менее 4 Гб;
- объем свободного места на жестком диске – 10 Гб.

Минимальные технические требования к серверу, на котором будет установлена СУБД с развернутой базой данных:

- оперативная память не менее 12 Гб;
- объем свободного места на жестком диске – 500 Гб.

2.2 Характеристика входной и выходной информации

Нормативно-справочная информация (НСИ) включает в себя словари, справочники, классификаторы, кодификаторы, нормативы и идентификаторы [12, 26].

В таблице 4 представлены реквизиты входных документов и их источники.

Таблица 4 – Характеристика входной информации

Входные данные	Объем в год	Источники получения
Регистрационные данные о поставщиках	10-15	Регистрационные данные поставщиков
Договор поставка	200-350	Бухгалтерия
Ассортимент ТМЦ	300-350	Каталог ТМЦ
Запрос на выдачу ТМЦ со склада	300-350	Зарегистрированная заявка на отгрузку
Акт перемещения товаров между складами	30-40	Отдел МТО
Акт списания ТМЦ	5-15	Отдел МТО

В информационное обеспечение входит база данных модуля информационной системы и входные документы.

В таблице 5 представлено описание выходных документов:

- выполненные поставки;
- невыполненные поставки;
- остатки на складах.

Таблица 5 – Описание выходных документов

Документ	Форма представления	Назначение
Выполненные поставки	Бумажный, электронный	Отдел МТО
Невыполненные поставки	Бумажный, электронный	Отдел МТО
Остатки на складах	Бумажный, электронный	Отдел МТО

На рисунке 14 представлен шаблон заказа поставщику.


ЗАКАЗ ПОСТАВЩИКУ № _____ от _____			
Поставщик: _____			
Покупатель: _____			
Срок исполнения: _____			
ИТОГО:			руб.
В том числе НДС:			руб.
Заказ оформил: _____			
Заказ отпустил: _____			
			

Рисунок 14 – Шаблон заказа поставщику

Шаблон требования-накладной на отгрузку товаров со склада представлен на рисунке 15.

Типовая межотраслевая форма № М-11 Утверждена постановлением Госкомстата России от 30.10.97 № 71а							
ТРЕБОВАНИЕ-НАКЛАДНАЯ № _____							Коды 0315006
Организация _____							Форма по ОКУД по ОКПО _____
Дата составления	Код вида операции	Отправитель		Получатель		Корреспондирующий счет	Учетная единица
		структурное подразделение	вид деятельности	структурное подразделение	вид деятельности	счет, субсчет	выпуска продукции (работ, услуг)
			цех		склад		
Через кого _____							
Затребовал _____				Разрешил _____			
Отпустил _____				Получил _____			
_____ (должность) _____ (подпись)		_____ (расшифровка подписи)		_____ (должность) _____ (подпись)		_____ (расшифровка подписи)	

Рисунок 15 – Шаблон требования-накладной

Выходные формы документов могут быть изменены пользователями уже после разработки и внедрения информационной системы.

2.3 Описание структуры базы данных

В процессе анализа были выделены основные информационные объекты, которые описывают процессы учета складских запасов на предприятии.

Для учета сведений об отделах используется сущность «Отдел», которая включает атрибуты «Код» и «Наименование».

Также в базе данных необходимо хранить информацию о сотрудниках: «Код», «ФИО», «Код отдела», «Должность», «Телефон» и «Электронная почта», «Дата рождения», «Дата начала работы».

Для исключения необходимости дублирования информации был выделен информационный объект «Должность» с атрибутами «Код» и «Наименование».

Для пополнения складских запасов заключается договора с поставщиками. Сведения о поставщиках также будут храниться в базе данных. Атрибутами информационного объекта «Поставщик» выбраны: «Код», «Наименование», «Адрес», «Телефон», «Электронная почта», «Контактные лица».

Материальные запасы поставляют на склад поставщики.

Выделена сущность «Поставка» с атрибутами «Код», «Код сотрудника», «Номер накладной», «Дата накладной», «Код поставщика», «Код склада».

Для описания количества поставленных товаров выделена сущность «Спецификация поставки» с атрибутами «Код», «Код поставки», «Код товара», «Количество», «Стоимость».

С каждым поставщиком может быть заключено несколько договоров, поэтому выделена сущность «Договор» с атрибутами «Код», «Номер договора», «Дата договора», «Код поставщика», «Дата начала», «Дата окончания», «Договор».

Сведения о материальных запасах хранятся в таблице «Товар» с атрибутами «Код», «Наименование», «Артикул», «Категория», «Характеристики».

Для учета категорий товаров используется сущность «Категория товаров» с атрибутами «Код», «Наименование».

Материальные запасы хранятся в складе. Для описания данного информационного объекта используется таблица «Склад» с атрибутами «Код», «Наименование», «Адрес», «Комментарии».

Специалист по снабжению оформляет заказы для поставки материальных запасов.

Сведения о заказах хранятся в таблице «Заказ». Атрибуты сущности: «Код», «Номер заказа», «Дата заказа», «Код поставщика», «Код сотрудника», «Дата выполнения».

Список товаров в заказе хранятся в таблице «Спецификация заказа» (атрибуты «Код», «Код заказа», «Количество», «Стоимость»).

История обработки заказов хранится в таблице «История обработки заказа» Атрибуты: «Код», «Код заказа», «Код статуса», «Код сотрудника», «Комментарии».

Для учета статусов заказов выделена сущность «Статус заказа» с атрибутами «Код», «Наименование».

Сведения о движении товаров между складами хранятся в таблице «Код», «Код товара», «Код движения», «Склад 1», «Склад 2», «Комментарии», «Дата движения», «Количество».

На рисунке 16 представлена ER-диаграмма базы данных в нотации Питера Чена [11, 13].

На рисунке 17 представлена логическая модель БД [19]. Физическая модель базы данных для СУБД MySQL представлена на рисунке 18.

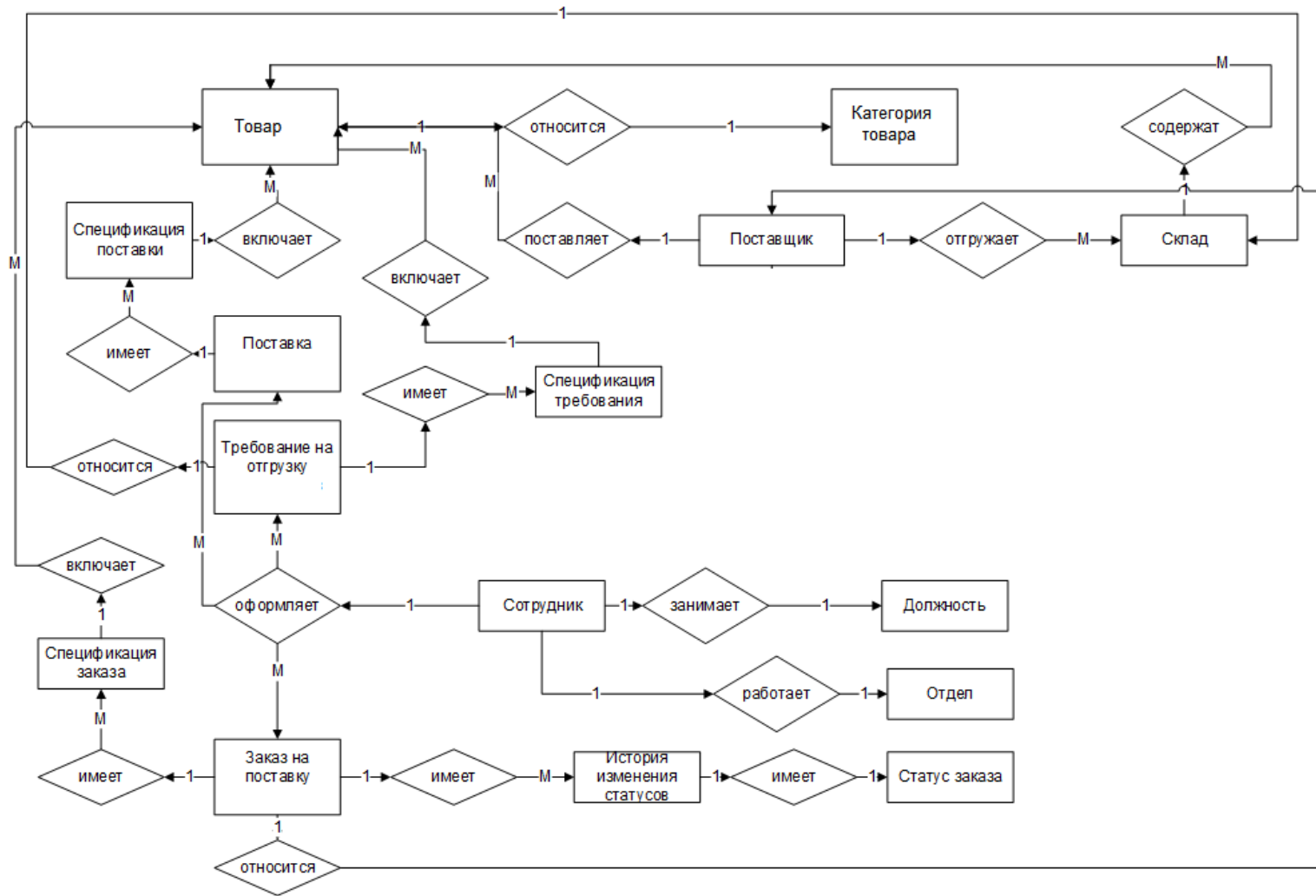


Рисунок 16 – ER-диаграмма

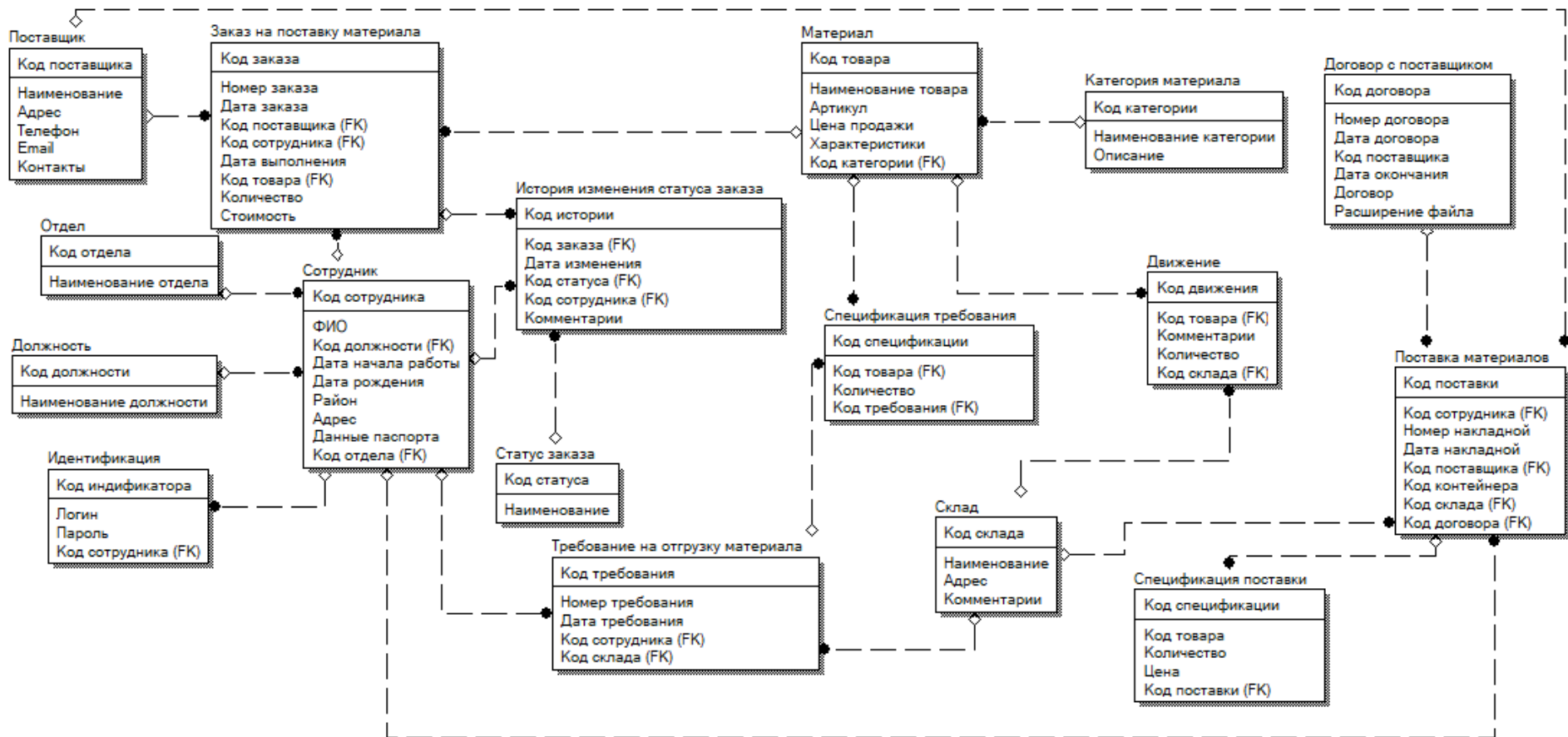


Рисунок 17 - Логическая модель БД

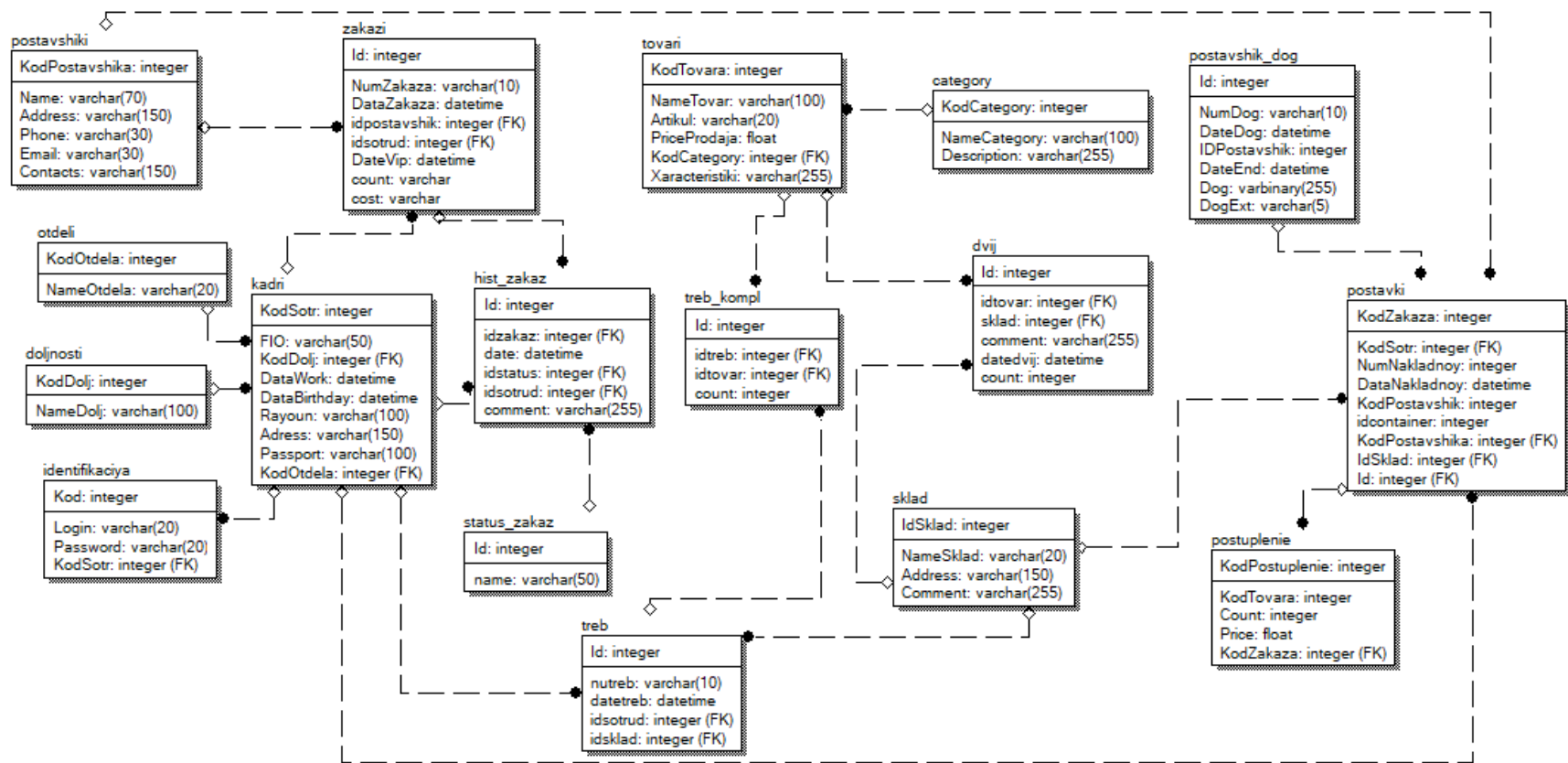


Рисунок 18 - Физическая структура базы данных

Физическая модель базы данных описывает типы данных для каждого атрибута.

Физическая модель была разработана с использованием программы AllFusion Data Modeler 7.3.

Выводы по второму разделу

Второй раздел работы включает описание технического и программного обеспечения, которое необходимо для решения поставленной задачи разработки информационной системы управления складскими запасами на предприятии. Дана характеристика входной и выходной информации, которая используется для проектирования и разработки структуры базы данных. Рассмотрена структура созданной базы данных.

3 Разработка информационной системы управления складскими запасами на предприятии

3.1 Дерево функций и описание интерфейса информационной системы управления запасами на предприятии

В соответствии с разработанными требованиями к информационной системе учета складских запасов было разработано дерево функций, представленное на рисунке 19.

Информационная система должна выполнять следующие функции:

- хранение и редактирование справочной информации;
- хранение и редактирование информации о сотрудниках организации;
- хранение и редактирование информации о материальных запасах;
- хранение и редактирование информации о поставках материальных запасов;
- хранение и редактирование информации о движении товаров между складами;
- автоматический расчет остатков товарно-материальных ценностей на складе - формирование документов: накладная на отгрузку товаров, заявка на отгрузку товаров;
- формирование отчетности: отчет о выполненных заказах, отчет о невыполненных заказах; отчет о поставке товаров поставщиками.



Рисунок 19 – Дерево функций информационной системы

Описание интерфейса информационной системы при авторизации в качестве специалиста по снабжению представлен на рисунке 20.

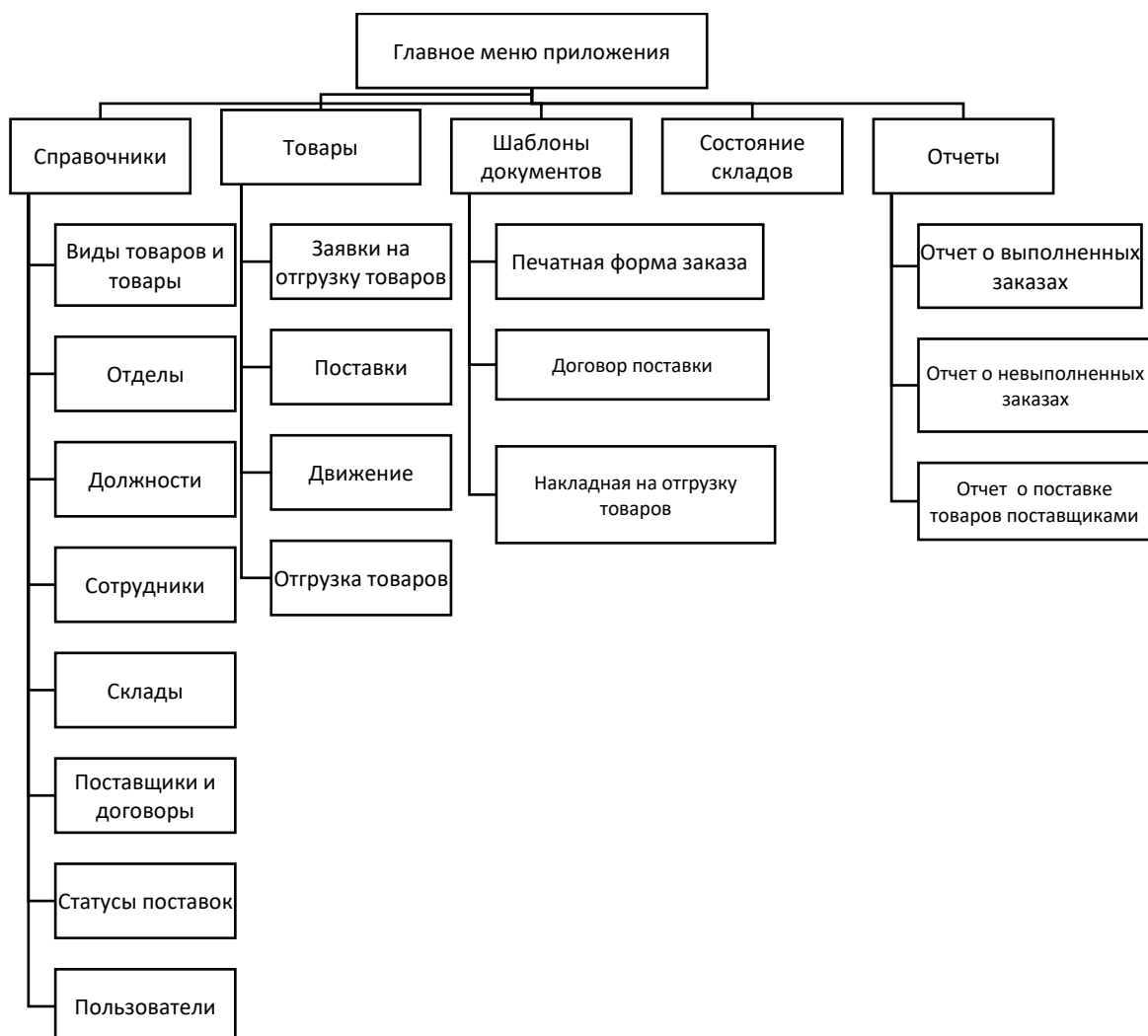


Рисунок 20 – Описание интерфейса информационной системы при авторизации в качестве специалиста по снабжению

Основные пункты меню:

- справочники;
- товары;
- шаблоны документов;
- состояние складов;
- отчеты.

3.2 Структурная схема разрабатываемой системы

Структура модуля информационной системы с указанием основных подсистем представлена на рисунке 21.



Рисунок 21 – Взаимосвязь подсистем модуля ИС

Для входа в систему пользователь вводит логин и пароль. Если авторизация прошла успешно, пользователь переходит к главной форме приложения.

Подсистема учета сотрудников предназначена для ведения учета сведений о сотрудниках организациях.

Подсистема учета поставщиков предназначена для:

- ведение учета поставщиков;
- ведение учета договоров с поставщиками;
- хранение копий договоров с поставщиками.

Подсистема учета поставок позволяет вести учет сведений о поставленных товарах на склад в рамках заключенных договоров с поставщиками. Для учета сведений о движении товаров между складами используется модуль учета движения товаров.

Подсистема формирования отчетности позволяет готовить отчетные материалы.

На рисунке 22 представлена схема работы информационной системы управления складскими запасами на предприятии ООО «СТД».

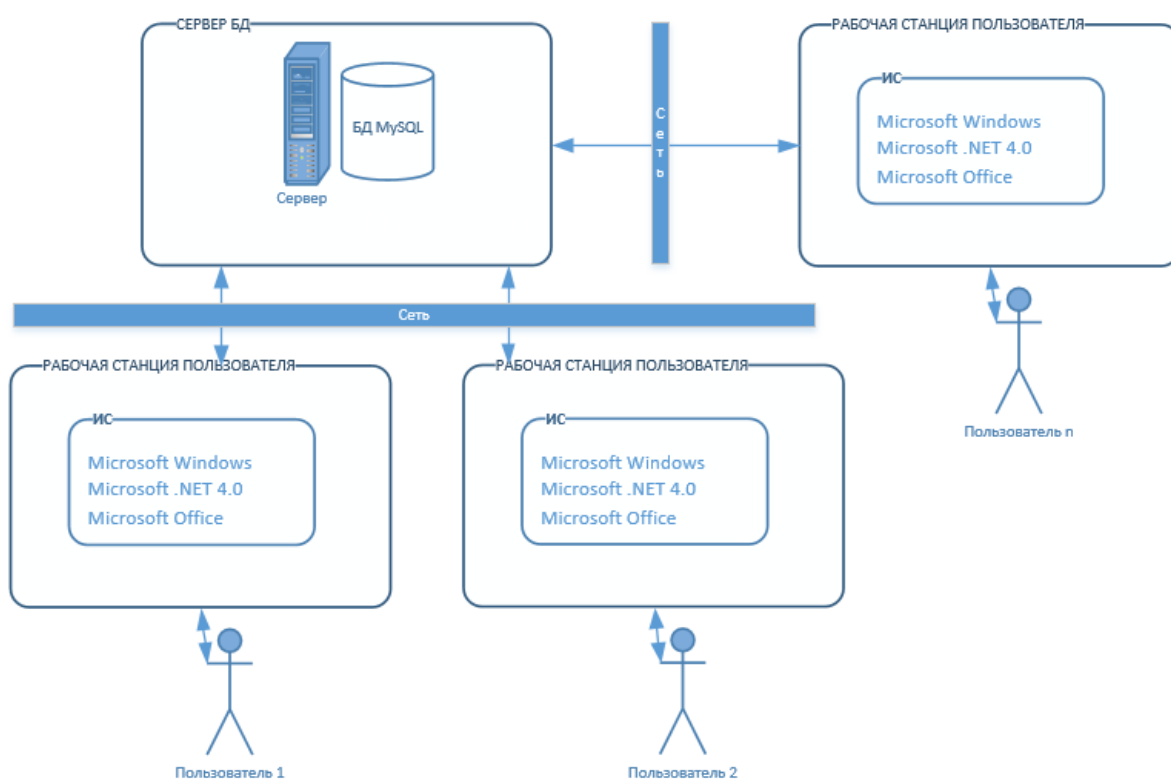


Рисунок 22 – Схема работы информационной системы управления складскими запасами

Для настройки работы с информационной системой необходимо разместить на сервере СУБД и выполнить с. На клиентских машинах устанавливается информационная система и настраивается доступ к серверу.

Так как модуль информационной системы разработан с помощью среды разработки приложений Microsoft Visual Studio 2019 необходимо наличие платформы .NET 4.8 на рабочих станциях пользователей.

3.3 Описание программных модулей

Информационная система учета материальных запасов предназначена для увеличения производительности работы специалистов по снабжению, занятых учетом и контролем складских запасов в организации.

Дистрибутив программы включает исполняемый файл Sklad.exe. Для развертывания БД необходимо в СУБД запустить на выполнение скрипт Sklad.sql.

Для настройки подключения к базе данных используются следующие настройки: `server=localhost; user id=root; password=root; persistsecurityinfo=True; database=sklad.`

Для запуска информационной системы необходимо запустить исполняемый файл Sklad.exe.

Доступ к данным БД реализован с помощью технологии ADO.NET.

В таблице 6 представлено описание разработанных программных модулей.

Таблица 6 – Назначение программных модулей

Наименование	Назначение модуля
1	2
Main	Основной модуль информационной системы, управление меню
FM_Auth	Проверка авторизации пользователя
FM_Container	Учет международных поставок в контейнерах

Продолжение таблицы 6

1	2
FM_Dolj	Справочник учета должностей
FM_Dvij	Редактирование информации о движении и списании товаров
FM_Kat	Редактирование видов категорий ТМЦ
FM_Otdeli	Редактирование отделов
FM_Polz	Редактирование пользователей
FM_Postav	Редактирование поставок товаров
FM_Sklad	Редактирование складов
FM_StatusZakaz	Редактирование статусов заказов
FromStatusContainer	Редактирование статусов контейнеров
FM_Reports	Формирование отчетов
FM_SostSklad	Остатки на складах
FM_Sotr	Редактирование сотрудников
FM_Tovari	Редактирование ТМЦ

На рисунке 23 представлен визуальный интерфейс системы администрирования базами данных MySQL, которая использовалась при разработке базы данных.

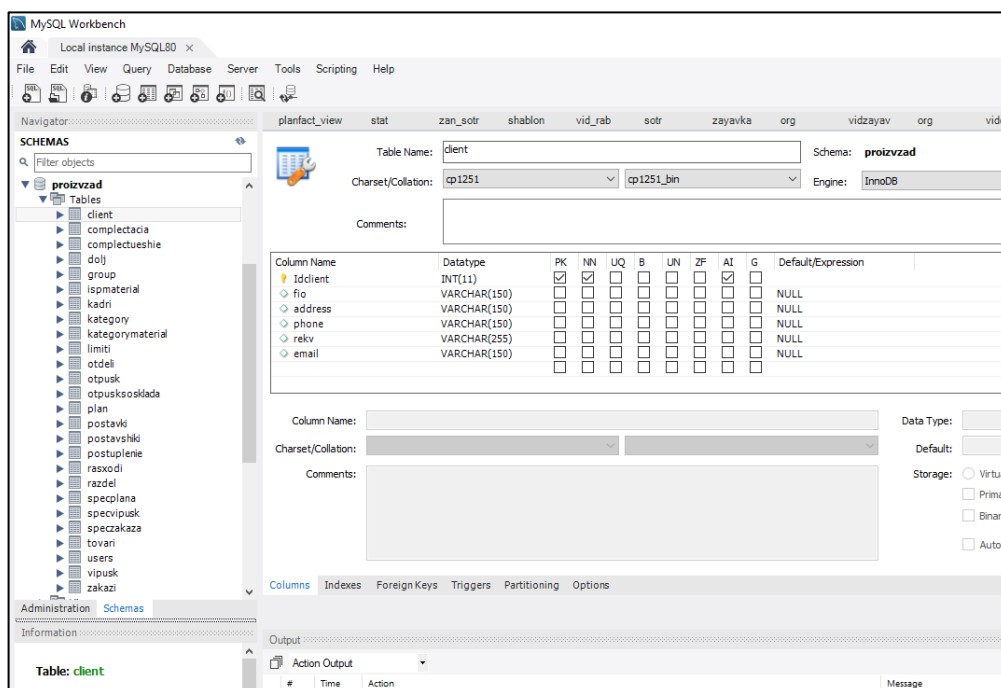


Рисунок 23 – Визуальный интерфейс MySQL Workbench

Диалоговое окно создания нового проекта в среде Visual Studio размещено на рисунке 24.

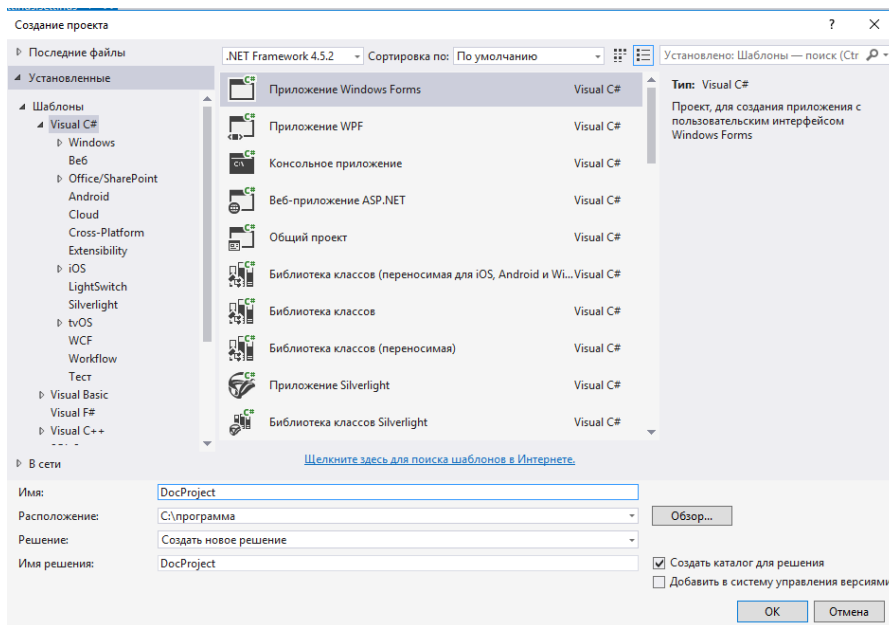


Рисунок 24 – Диалоговое окно создания нового проекта

Подключение проекта Visual Studio к базе данных выполнено с помощью добавления нового источника информации (рисунок 25).

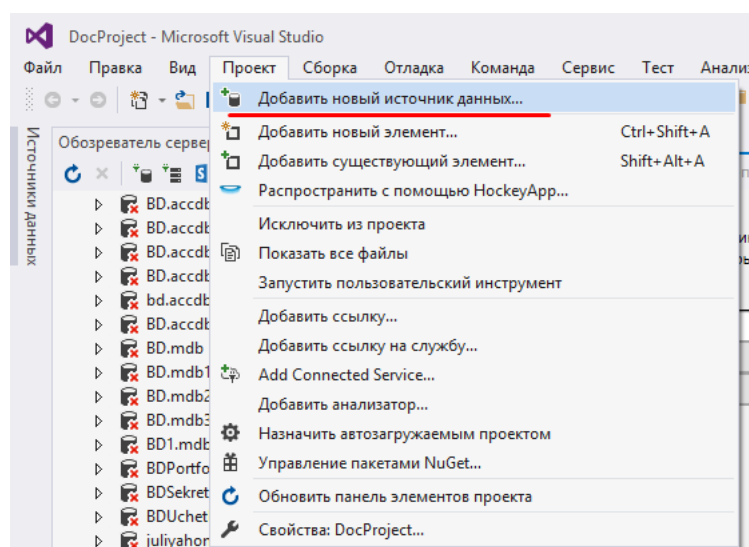


Рисунок 25 – Пункт меню «Проект» среды разработки Visual Studio

26. Диаграмма классов информационной системы представлена на рисунке

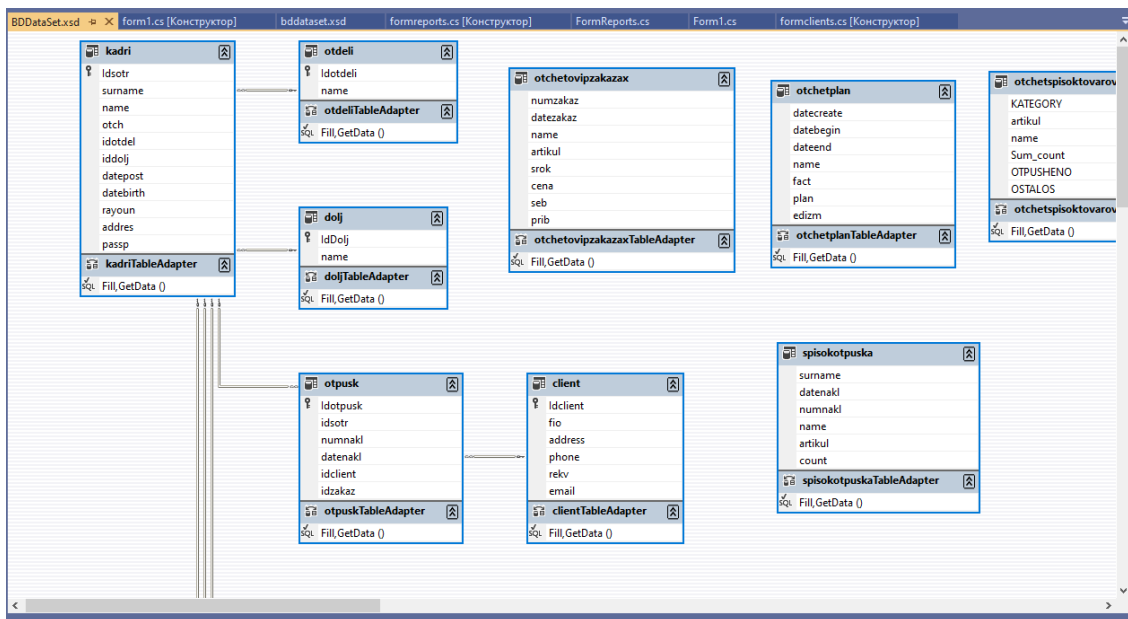


Рисунок 26 – Диаграмма классов MS Visual Studio

Пример формы редактирования должностей в режиме конструктора представлен на рисунке 27.

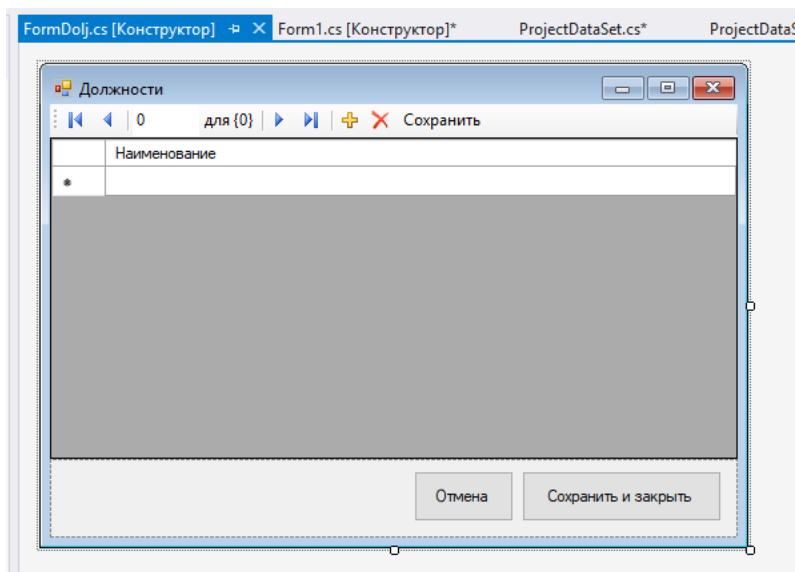


Рисунок 27 – Форма редактирования справочных данных

Пример подключения таблицы к источнику данных из базы данных MySQL представлен на рисунке 28.

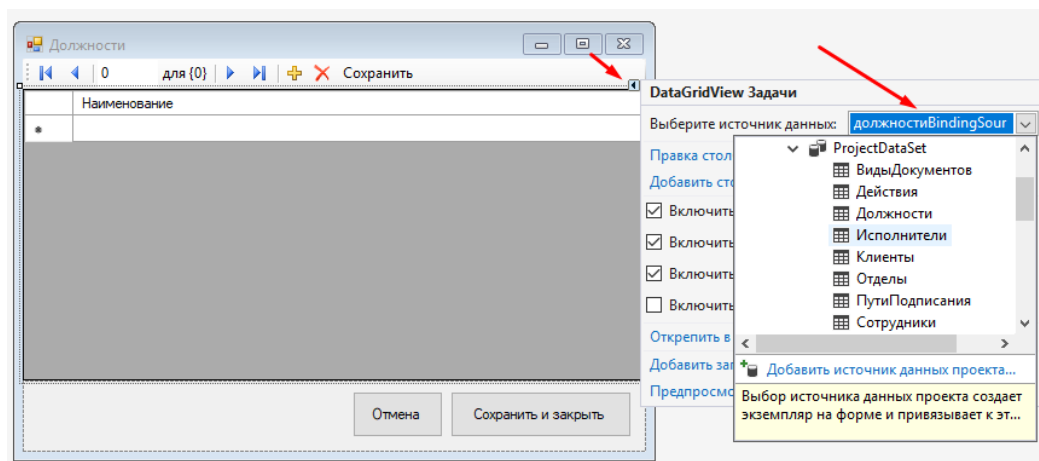


Рисунок 28 – Пример подключения таблицы к источнику данных из базы данных MySQL

Чтобы связать таблицу и объект базы данных, необходимо установить взаимную связь для данных элементов управления, как показано на рисунке 29.

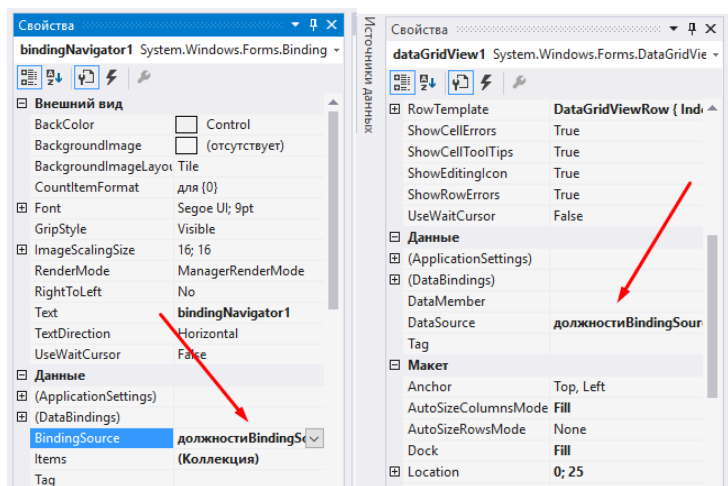


Рисунок 29 – Настройка подключения источника данных к таблице DataGridView

Также необходимо добавить программный код для обработки нажатия кнопки «Сохранить», чтобы обеспечить сохранение в базу данных информации, которая была изменена в таблице. Обрабатываются измененные, удаленные и обновленные данные.

Ниже представлен код обработки кнопки «Сохранить»:

```
this.Validate();
this.должностиBindingSource.EndEdit();
BDDDataSet.doljDataTable deletedOrders = (BDDDataSet.doljDataTable)
bDDDataSet.dolj.GetChanges(DataRowState.Deleted);

BDDDataSet.doljDataTable newOrders = (BDDDataSet.doljDataTable)
    bDDDataSet.dolj.GetChanges(DataRowState.Added);

BDDDataSet.doljDataTable modifiedOrders = (BDDDataSet.doljDataTable)
    bDDDataSet.dolj.GetChanges(DataRowState.Modified);

if (deletedOrders != null)
{
    должностиTableAdapter.Update(deletedOrders);
}
if (newOrders != null)
{
    должностиTableAdapter.Update(newOrders);
}
if (modifiedOrders != null)
{
    должностиTableAdapter.Update(modifiedOrders);
}
bDDDataSet.AcceptChanges();
this.должностиTableAdapter.Fill(this.bDDDataSet.dolj);
```

Подключение к базе данных указано в файле App.config.

Модуль контроля доступа выполняет проверку возможности работы с запрошенными данными авторизованному пользователю.

3.4 Схема работы информационной системы управления складскими запасами на предприятии

На рисунке 30 представлена схема работы информационной системы управления складскими запасами на предприятии, разработанная в соответствии с ГОСТ 19.701-90 «Схемы алгоритмов, программ, данных и систем».

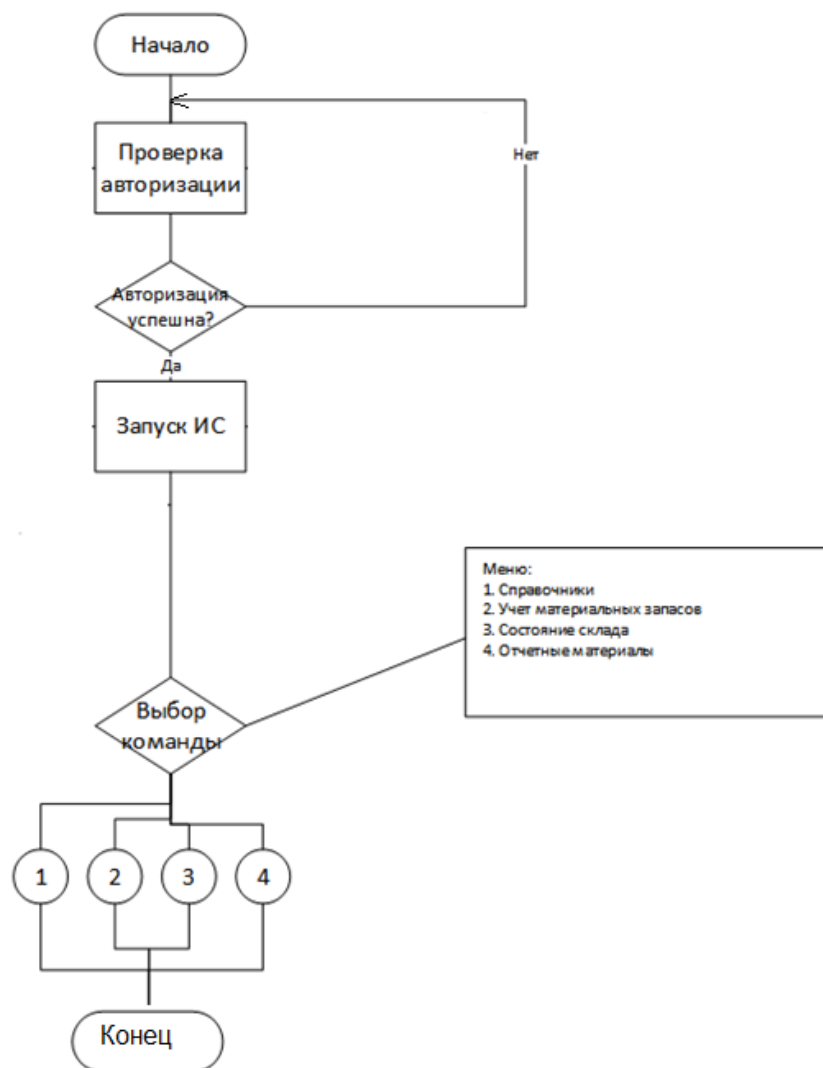


Рисунок 30 – Схема работы информационной системы управления складскими запасами на предприятии

При запуске системы пользователь должен ввести данные для проверки прав доступа в систему: логин и пароль. Если авторизация успешна, то будет открыта основная форма информационной системы, с помощью которой пользователь может:

- внести изменения в справочники;
- вести учета материальных запасов на складе;
- просматривать информацию об остатках товаров на складе.

3.5 Руководство пользователя

Работа с информационной системой начинается с обязательного ввода логина и пароля. Форма авторизации представлена на рисунке 31.

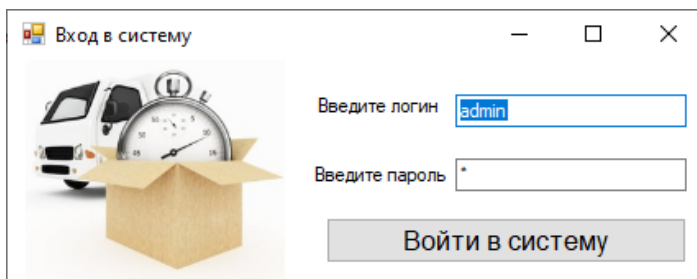


Рисунок 31 – Окно авторизации

При успешной авторизации появится главное окно приложения, представленное на рисунке 34.

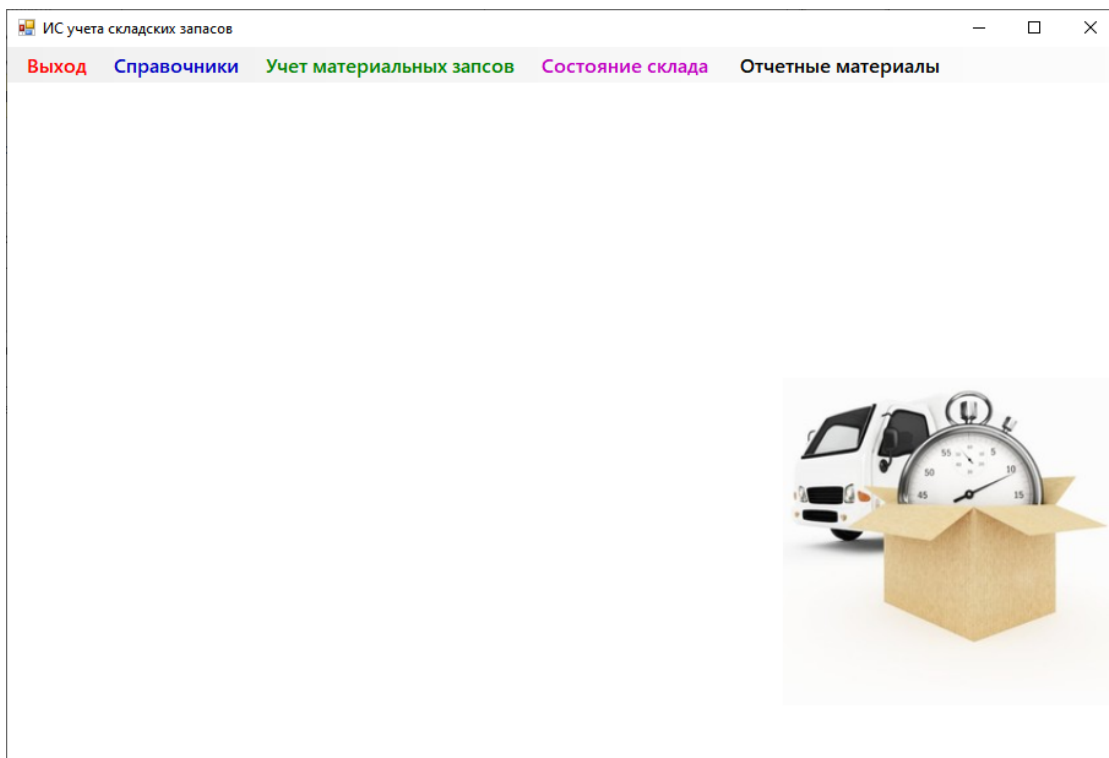


Рисунок 32 – Главное окно приложения

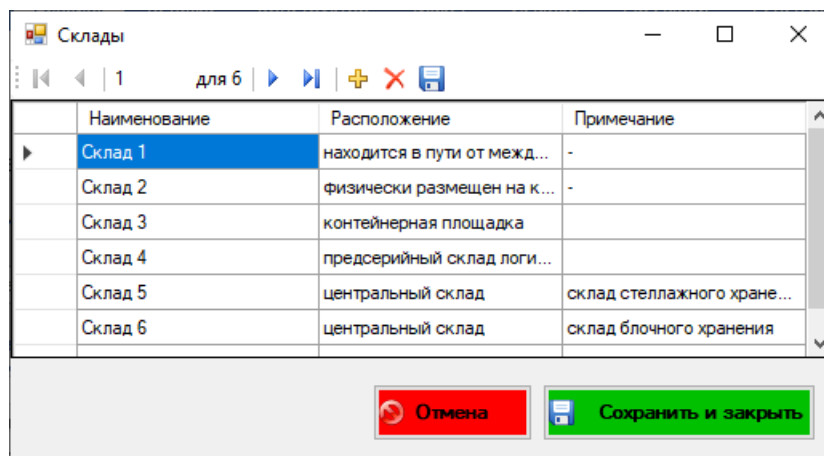
На рисунках 33-35 представлены скриншоты таблиц для редактирования справочных данных, используемых в разработанной системе.

Наименование	Описание
Реагенты	
Материалы	
Пропант	
Жидкость для гидроразрыва пласта	
Кислота для обработки призабойной зоны	
Реагенты для бурения	
Присадки	
Другое	
*	

Рисунок 33 – Скриншот таблицы «Категории товаров»

ФИО	Дата поступления в фирму	Дата рождения	Отдел	Должность	Район проживания	Адрес	Паспортные данные
Миненко А.П.	01.01.2011	08.12.1978	Бухгалтерия	бухгалтер	Пушкинский	ул.Пушкина 45	8569 856321 в...
Лазарева К.Е.	12.01.2012	18.05.1984	Отдел кадр...	специалист...	Ленинский	ул.Ленина 145 ...	7565 755665 в...
Пузанов П.Е.	11.08.2014	18.09.1997	Склад	грузчик	Железнодорож...	ул.Теневая 78	7854 256546 в...
Лемешева А.К.	12.07.2014	16.04.1987	Склад	заведующи...	Ленинский	ул.Войкова 74 ...	7896 125555 в...
Зими́на А.П.	01.05.2007	17.09.1974	Администр...	директор	Ленинский	ул.Ленина 147 ...	7896 655666 в...
*							

Рисунок 34 – Скриншот таблицы «Сотрудники»



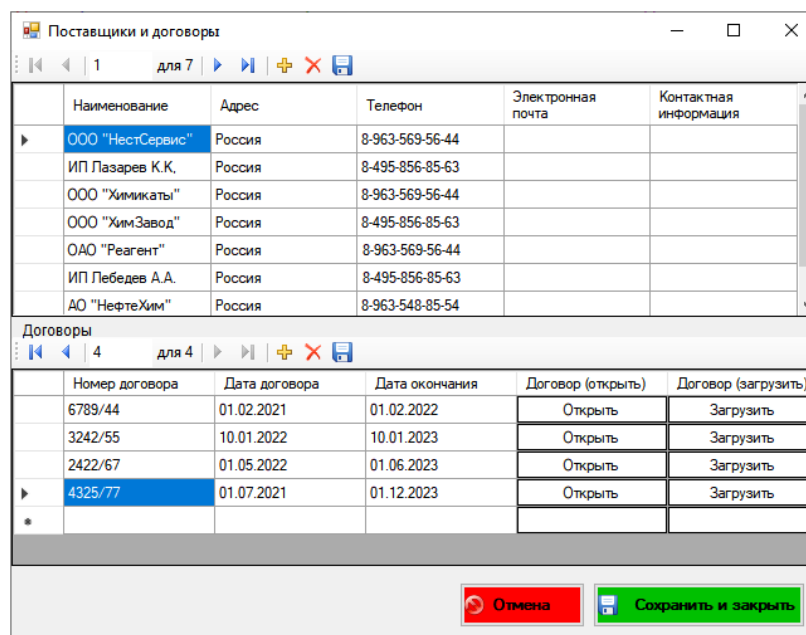
Наименование	Расположение	Примечание
Склад 1	находится в пути от межд...	-
Склад 2	физически размещен на к...	-
Склад 3	контейнерная площадка	
Склад 4	предсерийный склад логи...	
Склад 5	центральный склад	склад стеллажного хране...
Склад 6	центральный склад	склад блочного хранения

Рисунок 35 – Скриншот таблицы «Склады»

Компания заключает договоры с поставщиками на поставку реагентов и других материалов.

На рисунке 36 представлена форма «Поставщики и договоры».

Если в верхней части формы выбрать поставщика, то в нижней части будет доступен список заключенных с ним договоров. Пользователь может загрузить копию документа или открыть ранее загруженную для просмотра.



Наименование	Адрес	Телефон	Электронная почта	Контактная информация
ООО "НестСервис"	Россия	8-963-569-56-44		
ИП Пазарев К.К.	Россия	8-495-856-85-63		
ООО "Химикаты"	Россия	8-963-569-56-44		
ООО "ХимЗавод"	Россия	8-495-856-85-63		
ОАО "Реагент"	Россия	8-963-569-56-44		
ИП Лебедев А.А.	Россия	8-495-856-85-63		
АО "НефтеХим"	Россия	8-963-548-85-54		

Номер договора	Дата договора	Дата окончания	Договор (открыть)	Договор (загрузить)
6789/44	01.02.2021	01.02.2022	Открыть	Загрузить
3242/55	10.01.2022	10.01.2023	Открыть	Загрузить
2422/67	01.05.2022	01.06.2023	Открыть	Загрузить
4325/77	01.07.2021	01.12.2023	Открыть	Загрузить

Рисунок 36 – Скриншот таблицы «Поставщики и договоры»

На рисунке 37 представлена форма «Заказы на поставку», которая включает в себя полную информацию о заказах на поставку материалов, спецификации заказов и историю изменения заказов.

В верхней части формы находится список заказов.

При выборе заказа в нижней левой части формы отображается список товаров в заказе.

В нижней правой части – история изменения статусов заказов. Данная форма предназначена для просмотра информации и для ее редактирования.

История изменения заказа позволит отследить оперативности работы исполнителей в процессе последующего анализа деятельности всего отдела.

Номер заказа	Дата заказа	Поставщик	Сотрудник	Предв. дата закупки
1	09.04.2022	ООО "НестСервис"	Миненко А.П.	10.04.2022
2	10.04.2022	ООО "Химикаты"	Лазарева К.Е.	15.05.2022
3	11.04.2022	ИП Лазарев К.К.	Пузанов П.Е.	11.05.2022

Товар	Количество	Цена закупки, руб.
FY-89	100	80
GH-85-99	100	120
VFH90	100	250

Дата	Статус	Сотрудник	Комментарий
09.04.2022	новый	Миненк...	
10.04.2022	на согл...	Миненк...	
12.04.2022	подтвер...	Миненк...	
13.04.2022	включе...	Миненк...	

Оформить поставку в рамках выполнения договора: на склад

Оформить Печать заказа Закреть

Рисунок 37 – Форма «Заказы на поставку товаров»

На рисунке 38 представлена форма «Поставки материалов».

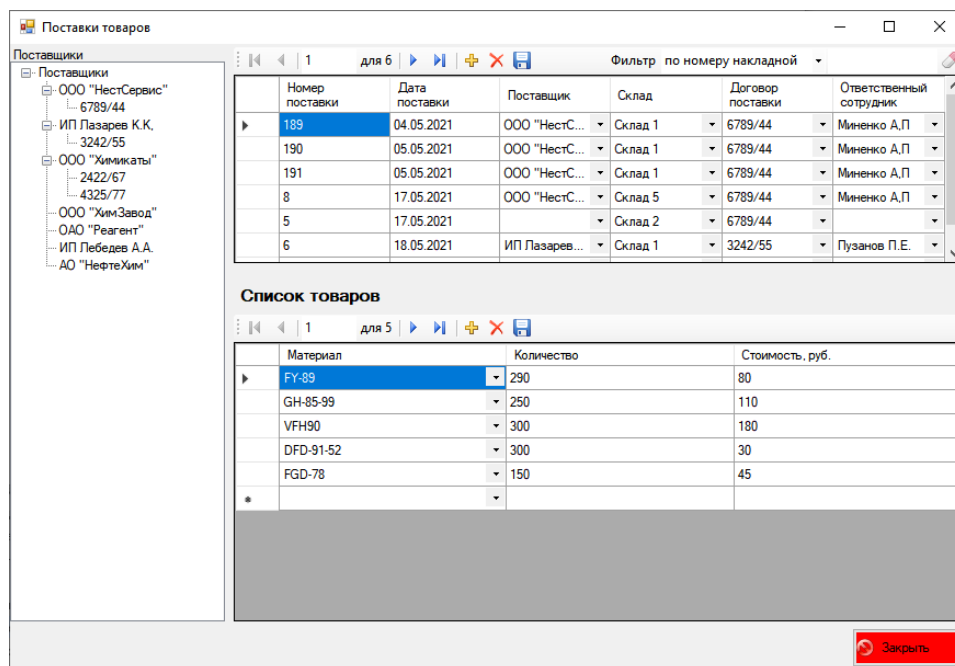


Рисунок 38 – Форма «Поставки товаров»

В левой части формы расположен список поставщиков и действующих договоров с поставщиками. При выборе поставщика в дереве на форме будут показаны только поставки выбранного поставщика. В нижней части формы расположен список материалов в поставке.

На рисунке 39 представлена форма «Движение материалов». Реализован фильтр по виду и периоду движения.

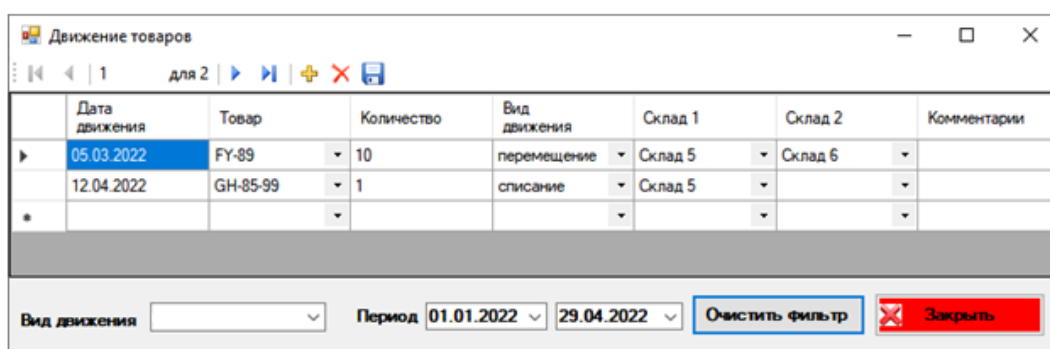


Рисунок 39 – Форма «Движение товаров»

На рисунке 40 представлена форма учета отгрузки товаров со склада.

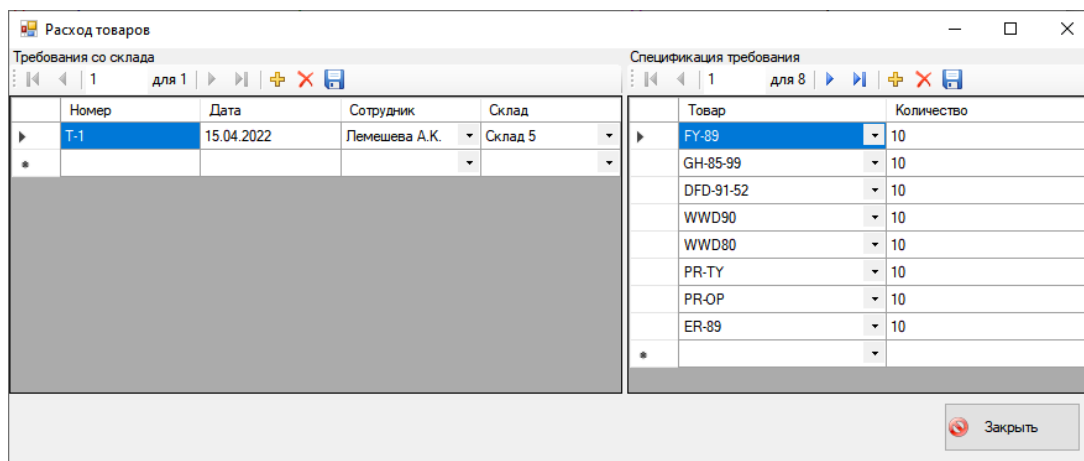


Рисунок 40 – Форма «Расход товаров»

На рисунке 41 представлена форма для просмотра информации о наличии товаров на складах организации.

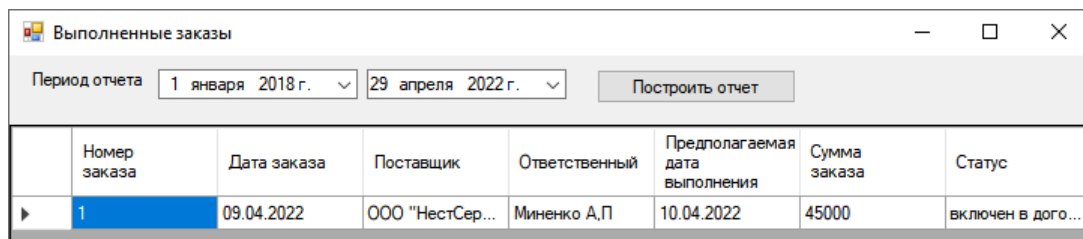
Наименование	Артикул	Категор	Склад	Поступило	Израсходов	Перемещен со склада	Перемещен на склад	Списано	Остаток
FGD-78	785566	Матер...	Склад 1	150					150
WWD90	232421	Матер...	Склад 1	200					200
PR-OP	342323	Проп...	Склад 1	300					300
PR-OI	243423	Проп...	Склад 1	200					200
PR-TY	243252	Проп...	Склад 1	400					400
PR-UY	324252	Проп...	Склад 1	350					350
RT7890	223422	Жидк...	Склад 1	400					400
ER-89	342563	Реаге...	Склад 1	450					450
ER-95	344244	Реаге...	Склад 1	300					300
GH-85-99	856667	Реаге...	Склад 2	100					100
FY-89	855856	Реаге...	Склад 2	100					100
VFH90	899966	Реаге...	Склад 2	100					100
FY-89	855856	Реаге...	Склад 5	100	10	10			80
GH-85-99	856667	Реаге...	Склад 5	100	10			1	89
VFH90	899966	Реаге...	Склад 5	100					100

Рисунок 41 – Информация о наличии товаров на складах организации

На данной форме можно увидеть общее количество израсходованных материалов, а также число списанных и перемещенных. Итоговая колонка содержит общее число оставшихся материалов.

Пользователь может воспользоваться фильтром или экспортировать данные в формат электронной таблицы.

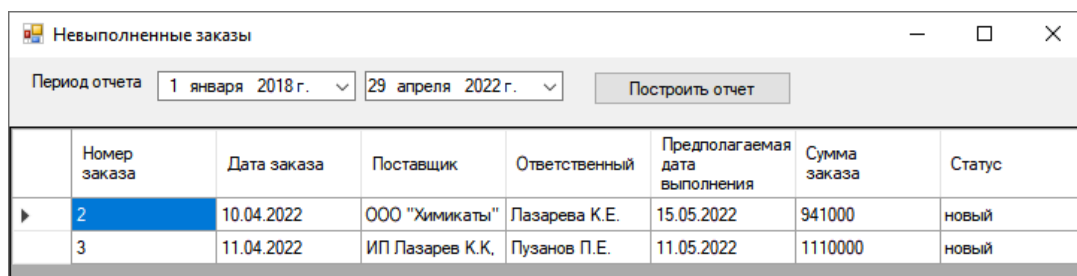
На рисунках 42 и 43 представлены примеры отчетов о выполненных и невыполненных заказах.



Скриншот окна «Выполненные заказы». Вверху указаны даты периода отчета: «1 января 2018 г.» и «29 апреля 2022 г.», кнопка «Построить отчет». Таблица содержит следующие данные:

Номер заказа	Дата заказа	Поставщик	Ответственный	Предполагаемая дата выполнения	Сумма заказа	Статус
1	09.04.2022	ООО "НестСер...	Миненко А,П	10.04.2022	45000	включен в дого...

Рисунок 42 – Пример отчета «Отчет о выполненных заказах»

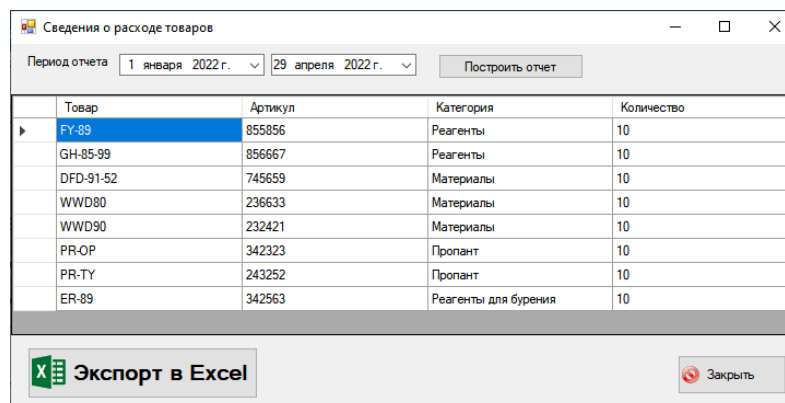


Скриншот окна «Невыполненные заказы». Вверху указаны даты периода отчета: «1 января 2018 г.» и «29 апреля 2022 г.», кнопка «Построить отчет». Таблица содержит следующие данные:

Номер заказа	Дата заказа	Поставщик	Ответственный	Предполагаемая дата выполнения	Сумма заказа	Статус
2	10.04.2022	ООО "Химикаты"	Лазарева К.Е.	15.05.2022	941000	новый
3	11.04.2022	ИП Лазарев К.К.	Пузанов П.Е.	11.05.2022	1110000	новый

Рисунок 43 – Пример отчета «Отчет о выполненных заказах»

На рисунке 44 представлен пример отчета «Отчет о расходе товаров».



Скриншот окна «Сведения о расходе товаров». Вверху указаны даты периода отчета: «1 января 2022 г.» и «29 апреля 2022 г.», кнопка «Построить отчет». Таблица содержит следующие данные:

Товар	Артикул	Категория	Количество
FY-89	855856	Реагенты	10
GH-85-99	856667	Реагенты	10
DFD-91-52	745659	Материалы	10
WWD80	236633	Материалы	10
WWD90	232421	Материалы	10
PR-OP	342323	Пропант	10
PR-TY	243252	Пропант	10
ER-89	342563	Реагенты для бурения	10

Внизу окна расположены кнопки «Экспорт в Excel» и «Закрыть».

Рисунок 44 – Пример отчета «Отчет о расходе товаров»

Отчетные формы и формы учета данных могут быть модифицированы и дополнены на этапе внедрения информационной системы.

При необходимости отчеты могут быть экспортированы в MS Excel.

Выводы по третьему разделу

В третьем разделе работы рассмотрено дерево функций, описан интерфейс информационной системы управления складскими запасами на предприятии. Описана структурная схема разработанной информационной системы управления складскими запасами на предприятии. Описаны программные модули, которые были разработаны для реализации проекта. Представлена схема взаимосвязи программных модулей. Представлено руководство пользователя информационной системы.

Заключение

Результатом выпускной квалификационной работы является информационная система управления складскими запасами для отдела материально-технического обеспечения, предназначенная для повышения эффективности работы сотрудников, занятых ведением учета материальных запасов.

Назначение информационной системы:

- хранение информации о поставщиках и заключенных договорах на поставку товаров;
- ведение учета перемещения товаров между складами предприятия, а также их списания;
- ведение учета поставок товаров;
- ведение учета ответственных сотрудников и их регистрационных данных для доступа к системе;
- формирование отчетов за указанный период о выполненных и невыполненных заказах на поставку материалов, а также о расходе товаров.

В первом разделе работы исследована деятельность сотрудников отдела материально-технического обеспечения по управлению складскими запасами. Рассмотрены основные функции и задачи отдела материально-технического обеспечения. Выполнено моделирование предметной области с помощью методологии IDEF0.

Целесообразность разработки информационной системы учета материальных запасов определяется следующими выводами:

- разработка и внедрение информационной системы позволит сократить количество ручных операций по ведению журналов учета поставок и расхода складских запасов;
- после внедрения информационной системы формирование отчетов будет выполняться без участия специалистов отдела МТО;

- исключение ручной обработки информации снизит вероятность погрешностей при анализе данных;
- после внедрения модуля информационной системы сократится количество бумажных документов.

Второй и третий раздел включают описание информационной модели. Представлен состав и структура классификаторов. Рассмотрены нормативно-справочные документы, реквизитный состав входных и выходных документов. Описана структура базы данных. Рассмотрен визуальный интерфейс информационной системы, представлена схема технологического процесса сбора информации.

Предполагается, что разработка информационной системы учета складских запасов позволит получить следующие результаты:

- сотрудники отдела МТО будут тратить меньше времени на заполнение журналов учета или вовсе избавятся от них;
- при учете данных с помощью модуля ИС будет контролироваться корректность внесенных данных;
- расчет остатков будет выполняться с помощью модуля ИС;
- упростить и ускорить процесс анализа данных;
- обеспечить оперативный поиск информации;
- обеспечить защиту информации.

В ходе выполнения выпускной квалификационной работы решены все поставленные задачи. Разработаны функциональные требования к справочно-информационной системе, требования к техническому, информационному и программному обеспечению.

В качестве СУБД выбрана MySQL, в качестве среды разработки выбрана современная система разработки приложений Microsoft Visual Studio на базе платформы .Net, язык программирования – C#.

Цель выпускной квалификационной работы достигнута.

Список используемой литературы

1. Барихин А.Б. Учет товарооборота: Учебник / А.Б. Барихин. – М.: Образование, 2019 – 194 с.
2. Гвоздева, Т.В. Проектирование информационных систем. Методы и средства функционального проектирования /Т.В. Гвоздева. М.: Лань, 2019. – 458 с.
3. Глотова, С.А. Информационные системы /С.А. Глотова. М.: Темика, 2020. – 494 с.
4. ГОСТ 34.003-90 Автоматизированные системы. Термины и определения.
5. ГОСТ 24.702-85 Единая система стандартов автоматизированных систем управления. Эффективность автоматизированных систем управления. Основные положения.
6. Гусов К. Н., Андреев А. А., Бондаренко Э. Н. Трудовое право России. Учебник для бакалавров. — М.: Проспект, 2021. 592 с.
7. Качанова Л. С. Моделирование бизнес-процессов на предприятиях АПК / Л.С. Качанова. – М.: Лань, 2020. – 172 с.
8. Лисяк В.В. Разработка информационных систем – М.: Дрофа, 2018 – 98 с.
9. Леонтьев В.П. Новейший самоучитель. Компьютер и интернет. СПб.: Питер, 2020. – 245 с.
10. Онлайн портал управления закупками AGORA [Электронный ресурс]. Режим доступа: <http://www.agora.htm> (дата обращения 14.04.2022).
11. Осипов Д.Л., Технологии проектирования баз данных / Д.Л. Осипов. СПб.: ДМК-Пресс, 2019, 498 с.
12. Официальный сайт «АИС «Комита» [Электронный ресурс]. URL: <http://www.komita.ru> (дата обращения 14.04.2022).
13. Официальный сайт «PROF-IT SRM: Управление закупками» [Электронный ресурс]. URL: <http://profitsrm.com/> (дата обращения 14.04.2022).

14. Семенова О.В. Информационные системы и программирование / О.В. Семенова. – М.: Инфра-М, 2020. – 214 с.
15. Тарасов С.В. СУБД для программиста. Базы данных изнутри / С.В. Тарасов.- М.: Библиотека профессионала, 2019. – 475 с.
16. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 03.07.2019).
17. Тумбинская Н.А., Защита информации на предприятии / Н.А. Тумбинская. М.:Лань, 2020. - 184 с.
18. Филимонова Е. В. Информатика и информационные технологии в профессиональной деятельности. Учебник. М.: Юстиция, 2019. 216 с.
19. Фомин Г.П. Математические методы в экономике. М.: Кнорус, 2019. – 328 с.
20. Хорев П.Б. Программно-аппаратная защита информации / П.Б. Хорев. – М.: Форум, 2019. – 352 с.
21. Худяков Г.А., Проектирование информационных систем / Г.А. Худяков. Учебное пособие, М.– Омега–Л, 2021. – 432 с.
22. Цацкина Е. П., Царегородцев А. В. Информатика и методы математического анализа. Учебно-методическое пособие. В 2 частях. Часть 1. Информатика. М.: Проспект, 2019. 96 с.,
23. Черников Б.В. Методология создания информационных систем / Б.В. Черников. М.; ИНФРА-М, 2021. – 320 с.
24. Чистов Д.В. Проектирование информационных систем: учебник и практикум.-М.:Юрайт, 2020.
25. Шаньгин, В.Ф. Информационная безопасность компьютерных систем и сетей: Учебное пособие / В.Ф. Шаньгин. - М.: Форум, 2018. - 256 с.
26. Шаймарданов, Р.Б. Моделирование и автоматизация проектирования структур баз данных / Р.Б. Шаймарданов. - М.: Радио и связь, 2020. – 120 с.
27. Шманев С.В. Прогнозирование и планирование экономики / Шманев С.В. — М.: Прометей, 2019. – 544 с.

28. Шмелева А. Г., Ладынин А. И. Информатика. Информационные технологии в профессиональной деятельности: Microsoft Word. Microsoft Excel: теория и применение для решения профессиональных задач. М.: ЛЕНАНД, 2020. 304 с.

29. Щербаков В.В. Автоматизация бизнес-процессов / В.В. Щербаков. М.: Питер, 2021. – 464 с.

30. Ansari Aness, Chell David. System DataBase. М.: Эком, 2013. – 944 с.

31. Lloyd S. Programming the Universe: A Quantum Computer Scientist Takes. Vintage, 2010. – 856 с.

32. Non-profit corporation founded to define transaction processing and database benchmarks and to disseminate objective, verifiable TPC performance data to the industry. [Электронный ресурс]. Режим доступа: <http://www.tpc.org>.

33. Sue Mosher. Microsoft Office Programming, Jumpstart for Administrators, Developers, and Power Users

34. Wolfgang Niefert. SAP Business One Implementation. Packt Publishing, 2009. – 112 с.

Приложение А Фрагменты исходного кода

```
namespace Gdo
{
    using System;
    using System.Collections;
    using System.IO;
    using System.Linq;
    using System.Text.RegularExpressions;
    using Aspose.Cells;
    using Fond;
    using Fond.Commands;
    using Fond.Documents;
    using Fond.Enums;
    using Fond.Helpers;
    using Fond.Icons;
    using Service.ConnectionHelper;
    using Service.Security;
    using TreeViewForBD.Versions;

    public class PluginAutoDispatcherReport : BaseExtendedPlugin<IViewPluginAutoDispatcherReport>
    {
        private CommandButton cmdBuildReport;
        private CommandDateTime cmdDateStart;
        private CommandDateTime cmdDateEnd;

        /// <summary>
        /// Будут вызывать этот метод другие плагины, для передачи этому плагину некоторого
        /// количества данных
        /// (Обязательно нужно подумать насчет помещения всего тела в lock {} (например, с
        /// помощью postSharp))
        /// </summary>
        /// <param name="message">передаваемый элемент</param>
        /// <param name="args">дополнительные параметры</param>
        /// <returns>true - элемент успешно обработан. Можно проверять, обработали ли этот элемент
        /// наследники плагина.</returns>
        public override void RecieveElement(IBaseMessage message, params object[] args)
        {
            this.IsFloatWindow = true;
            this.DockingMode = DockingStyleFondEnum.Float;
        }

        private bool CopyWorkSheet(Workbook workbook, int count, DateTime dateEnd)
        {
            var yearRgx = new Regex("\\d{4}");

            var targetSheets = workbook.Worksheets
                .Select(s =>
                {
                    var worksheet = s;

                    var name = worksheet.Name;

                    var month = FindMonth(name);
```

Продолжение Приложения А

```
if (month == -1) return null;

var match = yearRgx.Match(name);

if (match.Success == false) return null;

var date = new DateTime(int.Parse(match.Value), month, 1);

return new
{
    sheet = worksheet,
    date
};
})
.WhereNotNull()
.ToArray();

if (targetSheets.Length == 0) return false;

var dateSheetPair = targetSheets
    .FirstByMax(e => e.date);
var targetSheet = dateSheetPair.sheet;

for (int i = 0; i < count; i++)
{
    workbook.Worksheets.AddCopy(targetSheet.Index);
    var newWorkSheet = workbook.Worksheets.Last();
    newWorkSheet.Name = dateEnd.AddMonths(-i).ToString("MMMM yyyy");
    newWorkSheet.MoveTo(0);
}

workbook.Worksheets.RemoveAt(targetSheet.Index);

return true;

int FindMonth(string name)
{
    for (var i = 1; i < HelperDates.MonthsZa.Length; i++)
    {
        var monthStr = HelperDates.MonthsZa[i];
        if (name.ToLower().Contains(monthStr))
        {
            return i;
        }
    }

    return -1;
}

}

/// <summary>
/// Здесь инициализация плагина
/// </summary>
public override void PerformActionUser(object mainModule, params object[] args)
{
```

Продолжение Приложения А

```
this.IsFloatWindow = true;
    this.DockingMode = DockingStyleFondEnum.Float;
    base.PerformActionUser(mainModule, args);
    this.IsFloatWindow = true;
    this.DockingMode = DockingStyleFondEnum.Float;

    string holderName = Constants.CommandHolders.Default;
    ICommandHolder holderLocal = this.HolderDefault ?? this.View.CreateMenuHolder();

    this.cmdBuildReport = CommandBuilder.CommandButton("buildreport", "Построить отчет",
false, this.ExecutebuildReport);
    this.cmdBuildReport.IconLarge =
Icons.ИконкиПлагин.Нов_создатьДокумент(IconSizeEnum.Size24);

    this.cmdDateStart = CommandBuilder.CommandDateTime(null, "[Выбрать период]Дата
начала", false, null, null);
    this.cmdDateStart.Value = SessionContext.ContextProshloe.DateNow.AddYears(-
1).StartOfMonth();
    this.cmdDateStart.Icon =
Icons.ИконкиПлагин.Нов_показатьСкрытьПериоды(IconSizeEnum.Size16);
    this.cmdDateStart.IconLarge =
Icons.ИконкиПлагин.Нов_показатьСкрытьПериоды(IconSizeEnum.Size24);
    this.cmdDateStart.CommandDateTimeMode = CommandDateTimeMode.MonthYear;

    this.cmdDateEnd = CommandBuilder.CommandDateTime(null, "[Выбрать период]Дата
окончания", false, null, null);
    this.cmdDateEnd.Value = SessionContext.ContextProshloe.DateNow.StartOfMonth();
    this.cmdDateEnd.Icon =
Icons.ИконкиПлагин.Нов_показатьСкрытьПериоды(IconSizeEnum.Size16);
    this.cmdDateEnd.IconLarge =
Icons.ИконкиПлагин.Нов_показатьСкрытьПериоды(IconSizeEnum.Size24);
    this.cmdDateEnd.CommandDateTimeMode = CommandDateTimeMode.MonthYear;

    this.Commands.Add(holderName, this.cmdDateStart, this.cmdDateEnd, this.cmdBuildReport);
}

private void ExecutebuildReport(object sender, ExecuteEventArgs e)
{
    var templatePath =
ConnectionHelper.Instance.IService.GetExternalFileSource(Constants.ExternalFileSources.URL_TemplateDispatcherSut);
    if (string.IsNullOrEmpty(templatePath))
    {
        var errorText = $"Ошибка построения отчета.\r\n Не задана на сервере переменная
{Constants.ExternalFileSources.URL_TemplateDispatcherSut}. Отчет не может быть
построен.\r\n {Environment.UserName}\r\n {Environment.MachineName}";
        HelperEMail.SendEmailOverServer(ApplicationConstants.MailDeveloper, "Ошибка
построения отчета." + this.Caption, errorText, null);

        throw new ArgumentNullException(errorText);
    }

    if (!Directory.Exists(templatePath))
    {
```

Продолжение Приложения А

```
var errorText = $"Ошибка построения отчета.\r\n отсутствует папка
{ Constants.ExternalFileSources.URL_TemplateDispatcherSut}. Отчет не может быть
построен.\r\n{ Environment.UserName }\r\n{ Environment.MachineName }";
    HelperEMail.SendEmailOverServer(ApplicationConstants.MailDeveloper, "Ошибка
построения отчета." + this.Caption, errorText, null);

    throw new DirectoryNotFoundException(errorText);
}

var files = Directory.GetFiles(templatePath);

if (files.Length != 1)
{
    var errorText = $"В папке [{templatePath}] должен лежать только один файл шаблона для
формирования отчета. Разместите шаблон суточного диспетчерского отчета в эту
папку.\r\n{ Environment.UserName }\r\n{ Environment.MachineName }";
    HelperDialogs.alert(errorText, AlertMode.UserWarning);
    HelperEMail.SendEmailOverServer(ApplicationConstants.MailDeveloper, "Ошибка
построения отчета." + this.Caption, errorText, null);

    this.Close();
    return;
}

var excelFile = files[0];

var fileName = Path.GetFileName(excelFile);
//var ext = Path.GetExtension(excelFile);

var workFolder = ApplicationConstants.ApplicationDataFolder;
var newFileName =
DocumentManager.FileManager.GetUniqueFileNameForFolderWithoutFolder(Path.Combine(workFolde
r, fileName));
var dateStart = this.cmdDateStart.Value;
var dateEnd = this.cmdDateEnd.Value;
var monthCount = 0;

for (var date = dateStart; date <= dateEnd; date = date.AddMonths(1).StartOfMonth())
{
    monthCount++;
}

if (monthCount == 0)
{
    DocumentManager.FileManager.OpenFileInExternalApp(newFileName, out _);
}

using (var wb = new Workbook(excelFile))
{
    if (this.CopyWorkSheet(wb, monthCount, dateEnd))
    {
        wb.Save(newFileName);

        ApplicationContext.ThreadPool.QueueWorkItem(() =>
DocumentManager.FileManager.OpenFileInExplorer(newFileName));
    }
}
```

Продолжение Приложения А

```
        HelperDialogs.ShowDialogInformationReal($"Суточный отчет диспетчера {dateStart} -  
{dateEnd} сформирован.", "Отчет");  
        return;  
    }  
}  
  
return;  
  
        HelperDialogs.ShowDialogErrorReal($"Не найден шаблон автоматического расчета добычи",  
"Готово");  
  
    }  
}  
}  
  
using System;  
using System.Collections;  
using System.Collections.Generic;  
using System.Data;  
using Fond.Collections;  
using Fond.TreeDataSource;  
  
namespace Fond  
{  
    /// <summary>  
    /// Класс, визуализирующий структуру DataSet'a со всеми его связями (<see cref="DataRelation"/>)  
    /// </summary>  
    public partial class DataSetTree : ITreeDataSource  
    {  
        private readonly DataSet dataSource;  
  
        /// <summary>  
        /// Пустая коллекция для пустых коллекций дерева  
        /// </summary>  
        private readonly object[] emptyCollection = new object[] { };  
  
        /// <summary>  
        /// Список поддерживаемых колонок (FieldName, Caption).  
        /// !!! Реализовывать нужно через переменную, а не new ItreeDataSourceColumnInfo[] {}  
        /// </summary>  
        private IList<ItreeDataSourceColumnInfo> supportedColumns;  
  
        /// <summary>  
        /// Имя колонки, содержащей значение по умолчанию (не менять имя)  
        /// (Она невидима при отображении дерева, но значение можно получить)  
        /// </summary>  
        private const string columnThis = "this";  
  
        /// <summary>  
        /// Пример колонки  
        /// </summary>  
        private const string columnIdentifier = "Identifier";  
        private const string columnFieldName = "FieldName";  
  
        public DataSetTree(System.Data.DataSet dataSource)
```

```

{
    if (dataSource == null) throw new ArgumentNullException("dataSource");

    this.dataSource = dataSource;
}

/// <summary>
/// Корневой узел для заполнения виртуального дерева
/// </summary>
public object RootNode
{
    get { return "ROOT"; }
}

/// <summary>
/// Получить значение для определенной колонки дерева по ее fieldName
/// </summary>
/// <param name="nodeValue">Элемент, хранящийся в узле дерева(в данном случае
wrapper_aliasable_technical_reference)</param>
/// <param name="fieldName">fieldName у колонки</param>
/// <param name="handled">Говорит о том, что значение для колонки <paramref
name="fieldName"/> было получено. Иначе какой-то другой обработчик может дозаполнить поля и
нужно вернуть false</param>
/// <returns>Значение, присваиваемое ячейке в колонке с fieldName = <paramref
name="fieldName"/></returns>
object ITreeDataSource.GetCellValue(object nodeValue, string fieldName, out bool handled)
{
    switch (fieldName)
    {
        case columnThis:
            handled = true;
            return nodeValue;
        case columnIdentifier:
            //// здесь можно вернуть уже конкретное значение из nodeValue, например
            ((e_and_p_data)nodeValue).InstanceS
            handled = true;
            return GetIdentifier(nodeValue);
        case columnFieldName:
            handled = true;
            return GetName(nodeValue);
        default:
            handled = false;
            return nodeValue;
    }
}

private object GetIdentifier(object nodeValue)
{
    if (nodeValue is System.Data.DataTable)
    {
        return "[" + ((System.Data.DataTable)nodeValue).TableName + "]";
    }

    DataRelation rel = nodeValue as DataRelation;
    if (rel != null)

```

```

{
    if (IsRelationReverted(rel))
    {
        return "[" + ((System.Data.DataRelation)nodeValue).ParentTable.TableName + "];";
    }

    return "[" + ((System.Data.DataRelation)nodeValue).ChildTable.TableName + "];";
}

DataColumn column = nodeValue as DataColumn;
if (column != null)
{
    return string.IsNullOrEmpty(column.Caption) ? column.ColumnName : column.Caption;
}

return nodeValue;
}

private object GetName(object nodeValue)
{
    if (nodeValue is System.Data.DataTable)
    {
        return null;
    }

    DataRelation rel = nodeValue as DataRelation;
    if (rel != null)
    {
        string prefix = "DataRelation:";
        if (IsRelationReverted(rel))
        {
            prefix = "ОбратнаяDataRelation:";
        }

        return prefix + ((System.Data.DataRelation)nodeValue).RelationName;
    }

    DataColumn column = nodeValue as DataColumn;
    if (column != null)
    {
        return "ColumnName:" + column.ColumnName;
    }

    return nodeValue;
}
/// <summary>
/// Присвойте значение колонке дерева по ее fieldName
/// </summary>
/// <param name="nodeValue">Элемент, хранящийся в узле дерева(в данном случае
wrapper_aliasable_technical_reference)</param>
/// <param name="fieldName">fieldName у колонки</param>
/// <param name="value">Присваиваемое значение</param>
/// <param name="handled">Говорит о том, что значение для колонки <paramref
name="fieldName"/> было получено. Иначе какой-то другой обработчик может дозаполнить поля и
нужно вернуть false</param>

```

Продолжение Приложения А

```

/// <param name="errorText"></param>
/// <returns>Присвоено значение или нет</returns>
SetCellValueResultEnum ITreeDataSource.SetCellValue(object nodeValue, string fieldName, object
value, ref bool handled, ref string errorText)
{
    errorText = null;
    handled = false;
    return SetCellValueResultEnum.Ignore;
}

/// <summary>
/// Список поддерживаемых колонок (FieldName, Caption).
/// !!! Реализовывать нужно через переменную, а не new ItreeDataSourceColumnInfo[] {}
/// </summary>
IList<ItreeDataSourceColumnInfo> ITreeDataSource.SupportedColumns
{
    get
    {
        return this.supportedColumns ?? (this.supportedColumns = new ItreeDataSourceColumnInfo[]
        {
            new ItreeDataSourceColumnInfo(columnIdentifier, "Имя таблицы/колонки", true),
            new ItreeDataSourceColumnInfo(columnFieldName, "Внутреннее имя
таблицы/колонки/связи", true)
        });
    }
}

/// <summary>
/// Получить подчиненные узлы по parentу (используется для заполнения дерева)
/// </summary>
/// <param name="nodeValue">Элемент, хранящийся в узле дерева(в данном случае
wrapper_aliasable_technical_reference)</param>
/// <returns>Коллекция подчиненных объектов</returns>
IList ITreeData.GetChilds(object nodeValue)
{
    if (nodeValue == this.RootNode)
    {
        var result = new ArrayList(this.dataSource.Tables);
        result.Sort(ComparerDelegate<object>.GetComparer(UniversalComparer));
        return result;
    }

    if (nodeValue is System.Data.DataTable)
    {
        var result = new ArrayList(((System.Data.DataTable) nodeValue).Columns);
        result.Sort(ComparerDelegate<object>.GetComparer(UniversalComparer));
        return result;
    }

    if (nodeValue is System.Data.DataRelation)
    {
        DataRelation rel = (DataRelation) nodeValue;
        var result = new ArrayList(IsRelationReverted(rel)? rel.ParentTable.Columns :
rel.ChildTable.Columns);
        result.Sort(ComparerDelegate<object>.GetComparer(UniversalComparer));
    }
}

```



```

return result;
}

if (nodeValue is System.Data.DataColumn)
{
    var result = new ArrayList(GetRelationsForColumn((System.Data.DataColumn)nodeValue));
    result.Sort(ComparerDelegate<object>.GetComparer(UniversalComparer));
    return result;
}

return this.emptyCollection;
}

private int UniversalComparer(object x, object y)
{
    if ((x is DataTable) && !(y is DataTable))
    {
        return 0.CompareTo(1);
    }

    if (!(x is DataTable) && (y is DataTable))
    {
        return 1.CompareTo(0);
    }

    if ((x is DataTable) && (y is DataTable))
    {
        return String.CompareOrdinal(((DataTable) x).TableName, ((DataTable) y).TableName);
    }

    if (x is DataColumn && y is DataColumn)
    {
        return String.CompareOrdinal(((DataColumn) x).Caption, ((DataColumn) y).Caption);
    }

    if (x is DataRelation && y is DataRelation)
    {
        return String.CompareOrdinal(((DataRelation)x).ChildTable.TableName,
        ((DataRelation)y).ChildTable.TableName);
    }

    return System.String.Compare(Convert.ToString(x), Convert.ToString(y),
    System.StringComparison.Ordinal);
}

private bool IsRelationReverted(DataRelation relation)
{
    object revert = relation.ExtendedProperties["revert"];
    if (revert == null) return false;
    return (bool) revert;
}

private IList GetRelationsForColumn(DataColumn column)
{
    if (column == null) return this.emptyCollection;
}

```

Продолжение Приложения А

```

ArrayList result = new ArrayList();
foreach (System.Data.DataRelation childRelation in column.Table.ChildRelations)
{
    if (Equals(childRelation.ParentColumns[0], column))
    {
        result.Add(childRelation);
    }
}

//if (result.Count == 0)
{
    /// обратные связи найдем
    foreach (DataRelation dataRelation in column.Table.ParentRelations)
    {
        if (Equals(dataRelation.ChildColumns[0], column))
        {
            DataRelation revert = new
DataRelation(dataRelation.RelationName,dataRelation.ParentColumns[0],dataRelation.ChildColumns[0])
;
            revert.ExtendedProperties.Add("revert",true);
            result.Add(revert);
        }
    }
    //foreach (System.Data.DataRelation parentRelation in column.Table.ParentRelations)
    //{
    //    if (Equals(parentRelation.ChildColumns[0], column))
    //    {
    //        result.Add(parentRelation);
    //    }
    //}
}
return result;
}

/// <summary>
/// Возвращает данные, которые ассоциированным с узлом дерева
/// </summary>
/// <param name="node">Узел дерева для получения данных</param>
/// <returns>Данные хрянящиеся в узле дерева</returns>
object ITreeData.GetNodeData(object node)
{
    return GetIdentifier(node);
}
}
}

```