

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»
Кафедра IT – инжиниринг

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой PhD, доцент Досжанова А.А
_____ « ____ » _____ 2020 г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка системы учета движения товаров на складе малого предприятия

Специальность 5В070400 – Вычислительная техника и программное обеспечение

Выполнила: Нурланулы М. Группа ВТ-16-2

Научный руководитель: PhD, доцент Досжанова А.А

Консультанты:

по экономической части: к.э.н., профессор Габелашвили К.Р

(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

по безопасности жизнедеятельности: к.т.н., доцент Приходько Н.Г

(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

по программному обеспечению: ст.преп. Майкотов М.Н

(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

Нормоконтролер: ст.преп. Абсатарова Б.Р

(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

Рецензент: _____

(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В070400 – «Вычислительная техника и программное обеспечение»

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Нурланулы Мади

Тема проекта: Разработка системы учета движения товаров на складе малого предприятия

Утверждена приказом по университету № ____ от «____» _____ 2020 г.

Срок сдачи законченного проекта «____» _____ 2020 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): руководство по программированию на 1С, управление реляционными базами данных, данные преддипломной практики.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

- а) анализ и исследование предметной области;
- б) проектирование приложения;
- в) реализация приложения;
- г) экономическое обоснование эффективности;
- д) вопросы безопасности жизнедеятельности и охраны труда.

Перечень графического материала (с точным указанием обязательных чертежей): имеется 25 таблиц и 40 иллюстраций.

Основная рекомендуемая литература:

1 1С:Предприятие 8.3. Практическое пособие разработчика /Авторы: Радченко Максим Григорьевич, Хрусталева Елена Юрьевна. , 2017 г.

2 Решение специальных прикладных задач в «1С:Предприятии 8.2» /Авторы: Гончаров Д.И., Хрусталева Е.Ю, 2012 г.

3 Профессиональная разработка в системе 1С:Предприятие 8.3" / Авторы: Ажеронок В.А., Габец А.П., Гончаров Д.И., Козырев Д.В., Кухлевский Д.С., Островерх А.В., Радченко М.Г., Хрусталева Е.Ю. , 2012 г.

4 Разработка управляемого интерфейса /Авторы: Ажеронок В.А., Островерх А. В., Радченко М. Г., Хрусталева Е. Ю. , 2010 г.

5 Разработка сложных отчетов в «1С:Предприятии 8.3». Система компоновки данных /Авторы: Хрусталева Е.Ю. , 2012 г.

Консультация по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Габелашвили К.Р	24.04.2020	
Безопасности жизнедеятельности	Приходько Н.Г	24.04.2020	
Программная часть	Майкотов М.Н	14.05.2020	
Нормконтролер	Абсатарова Б.Р	18.05.2020	

ГРАФИК
подготовки дипломной работы (проекта)

Наименования разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Анализ и исследование предметной области	01.11.2019– 20.12.2019	
Проектирование приложения	21.12.2019 – 20.02.2020	
Программная реализация	21.02.2020 – 20.04.2020	

Дата выдачи задания « ____ » _____ 2020 г.

Заведующий кафедрой _____ А.А.Досжанова
(подпись)

Научный руководитель проекта _____ А.А.Досжанова
(подпись)

Задание принял к исполнению студент _____ М. Нурланулы
(подпись)

Аңдатпа

Дипломдық жобаның тақырыбы: «Шағын кәсіпорынның қоймасындағы тауарлардың қозғалысын есепке алу жүйесін жасау».

Бұл дипломдық жоба шағын кәсіпорынның қоймасындағы тауарлардың қозғалысын есепке алу жүйесін құруды жүзеге асырды, оның мақсаты тауарлардың қозғалысын есепке алудың автоматтандырылған кешенін құру болып табылады.

Дипломдық жоба 1С қосымшасы форматында орындалды. Дайындалған дипломдық жобаның мақсатына жету үшін Visual Basic тіліне ұқсас интеграцияланған 1С тілі сияқты технологиялар қолданылды.

Бағдарламаны құру жұмсалған шығындар мен шығындардың экономикалық есебі, құрылған жобаның экономикалық мақсаттылығы бағаланды.

Аннотация

Тема дипломного проекта: «Разработка системы учета движения товаров на складе малого предприятия».

В данном дипломном проекте реализована разработка системы учета движения товаров на складе малого предприятия, цель которого состоит в том, чтобы реализовать автоматизированный комплекс учета движения товаров.

Дипломный проект был выполнен в формате приложения 1С. Для достижения цели разрабатываемого дипломного проекта были применены такие технологии, как встроенный язык 1С, в качестве подобного можно рассмотреть язык Visual Basic.

Также был проведен экономический расчет затрат и стоимости разработки программы, оценка экономической целесообразности разрабатываемого проекта.

Abstract

The theme of the graduation project: "Development of a system for recording the movement of goods in a warehouse of a small enterprise."

This graduation project has implemented the development of a system for recording the movement of goods in a warehouse of a small enterprise, the purpose of which is to implement an automated complex for recording the movement of goods.

The graduation project was performed in the format of the application 1C. To achieve the goal of the developed graduation project, technologies such as the integrated 1C language were applied, as a similar language Visual Basic can be considered.

An economic calculation of the costs and costs of developing the program, an assessment of the economic feasibility of the developed project was also carried out.

Содержание

Введение	8
1 Аналитическая часть	Ошибка! Закладка не определена.
1.1 Теоретический анализ предметной области	Ошибка! Закладка не определена.0
1.2. Обзор аналогов	Ошибка! Закладка не определена.1
1.3. Постановка задач	Ошибка! Закладка не определена.5
2. Проектирование программного продукта	Ошибка! Закладка не определена.6
2.1 Структура программного обеспечения	Ошибка! Закладка не определена.6
2.2. Обоснование выбора средств	Ошибка! Закладка не определена.6
3 Разработка программного комплекса	Ошибка! Закладка не определена.3
3.1 Создание кода	Ошибка! Закладка не определена.4
3.2 Системные требования	Ошибка! Закладка не определена.
3.3. Разработка экранных форм ...	Ошибка! Закладка не определена.1
4 Технико-экономическое обоснование	Ошибка! Закладка не определена.0
4.1 Определение сложности разработки ПО	Ошибка! Закладка не определена.0
4.2 Расчет затрат на разработку ПО	Ошибка! Закладка не определена.1
4.3 Расчет затрат на электроэнергию	Ошибка! Закладка не определена.3
4.4 Расчет затрат на оплату труда	Ошибка! Закладка не определена.4
4.5 Расчет затрат по социальному налогу	Ошибка! Закладка не определена.5
4.6 Амортизация основных фондов и прочие затраты	Ошибка! Закладка не определена.6
4.7 Определение возможной (договорной) цены ПО	Ошибка! Закладка не определена.7
4.8 Сравнительный анализ эксплуатационных затрат до и после внедрения	Ошибка! Закладка не определена.8
4.9 Вывод по экономической части	Ошибка! Закладка не определена.2
5. Безопасность жизнедеятельности	Ошибка! Закладка не определена.3
5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал	Ошибка! Закладка не определена.3
5.2 Расчет естественного освещения	Ошибка! Закладка не определена.5
5.3 Расчет систем пожаротушения	Ошибка! Закладка не определена.7

5.4 Расчет систем кондиционирования рабочего помещения складского персонала.....	89
Заключение	Ошибка! Закладка не определена. 2
Список литературы	Ошибка! Закладка не определена. 3
Приложение А (Техническое задание).....	Ошибка! Закладка не определена. 5
Приложение Б (Листинг программы).....	98
Приложение В (Акт внедрения) ...	Ошибка! Закладка не определена. 08

Введение

В наши дни всё большее количество субъектов бизнеса стремятся автоматизировать рабочий процесс с использованием информационных технологий. На данный момент хранение, поиск и доступ к информации стали неотъемлемой частью не только жизни людей, тесно связанных с деятельностью в сфере компьютерных технологий, но и входит в работу других сфер социума. Данное решение является способом сокращения временных, материальных и даже физических затрат на поиск, приобретение, хранение, обмен различными материальными и нематериальными ресурсами.

Достижение максимальных выгод от перехода страны на рыночные отношения и признания ее в качестве рыночно ориентированного промышленно развитого государства требуют кардинальных преобразований в нормативной базе управления деятельностью предприятий, включая создание научно обоснованной системы нормативного регулирования бухгалтерского учета. Однако недостатки действующей системы нормативного регулирования бухгалтерского учета и создаваемые ею барьеры препятствуют формированию стратегически и тактически обоснованного учетно-информационного обеспечения управления деятельностью предприятий, поиску и мобилизации резервов повышения эффективности предпринимательства, обеспечению выживаемости и развития организаций в условиях жесткой конкурентной борьбы, коммерческой тайны, инфляции и банкротства.

Торговля является важнейшей отраслью хозяйственной деятельности. В торговой сфере происходит движение товарных и финансовых течений, которые являются неотъемлемой частью общества. Для обеспечения успешного управления деятельностью руководству как воздух необходима полная, точная, объективная и своевременно поданная детальная экономическая информация. И в данной сфере склад является неотъемлемой и несомненно одной из важнейших частей предприятия.

Склад принимает партии товаров от поставщиков и отпускает его клиентам мелкими партиями. Продукция поступает непосредственно с заводов-изготовителей и путем реализации попадает к клиентам, имеющим нужду в данном виде товаров. Предприятие осуществляет передвижение продукции с производства на базу, ее складской учет, хранение и отгрузку клиентам.

Требуется вести количественный и стоимостной учет таких составляющих, как, товары: поступающие и отпускаемые, поставщиков и клиентов, формировать отчетности первичной внутренней и бухгалтерской документации.

Основными задачами складского учета являются:

- обеспечение контроля за движением и правильным использованием всех складских товаров и ценностей;
- соблюдение установленных критериев запасов и расходов;
- своевременное выявление неиспользуемых товарных составляющих, подлежащих реализации или утилизации в требуемые сроки;
- создание отчетных материалов с требуемой для этого аналитикой.

Актуальность выбранной темы обусловлена тем, что в первую очередь для успешного развития любого предприятия, в независимости от типа предприятия и деятельности, что данная компания осуществляет, необходимо правильно организовать работу его склада. Для данных предприятий требуется помещение, оборудованное под склад и охраняемое от сторонних посягательств на товар и другое содержимое складского помещения. Сохранность всего объема товарных и материальных ценностей предприятия является одним из основных плюсов в пользу использования складского помещения для предприятий малого и крупного бизнеса.

Цель дипломного проекта - на основе изучения движения учета товаров в организациях малой торговли, результатов анализа системы учета на исследуемом объекте, определить основные направления совершенствования учета движения товаров и реализовать комплекс учета движения товаров, включающим в себя ведение отчетности первичной внутренней и бухгалтерской документации.

1 Аналитическая часть

В данной главе необходимо рассмотреть задачи, которые будет выполнять разрабатываемая информационная система, изучить аналоги и прототипы, выявить их достоинства и недостатки. А также выбрать СУБД, среду программирования для создания ИС и операционную систему.

1.1 Теоретический анализ предметной области

Улучшение уровня управления системой и оперативности принятия решений являются основополагающими решениями в вопросе повышения эффективности работы систем управления складским учетом.

К показателям эффективности относятся:

- актуальность данных и скорость их обновления;
- время, затрачиваемое на принятие решений и решение задач, в том числе задач управленческого характера.
- качество информации, ее достоверность, детальность;
- ускорение подготовки документов, уменьшение временных затрат на выходные документы;
- повышение качества труда за счет сокращения рутинных операций.

Предстоит дать оценку эффективности применения нашей системы используя решения виду прямых и косвенных показателей оценки. Экономические, также именуемые прямыми, показатели позволяют получить оценку автоматизации в денежном эквиваленте, включают в себя оценку расходов на процесс разработки и работы системы учета (полная стоимость всей системы складского учета), определение входного денежного состава, высвобожденного за счет использования системы. Косвенные же показатели являются направленным инструментом, позволяющим оценить пользу от ИС в выбранной предметной области.

Эффективность комплекса имеет экономический эффект от внедрения в работу организации и функционирования ИС, и также такие составляющие, как высокая надежность, легкость в эксплуатации, возможность модификации, уменьшение затрат на разработку и эксплуатацию.

Система учета движения товаров на складе - информационная система, осуществляющая передачу оперативных данных, требуемых персоналу для успешного управления складом предприятия в режиме реального времени.

В ИС заложен широкий функционал для учета товара и других задач складского учета. В то же время программа понятна в использовании и не является сложным инструментом для непрофессионального пользователя в ней легко работать и ориентироваться.

Комплекс предоставляет и обеспечивает работающего широким списком обобщающих и аналитических отчетов, позволяющих быстро и

просто найти любую интересующую информацию в простой для понимания экранной форме.

Функционал программы позволяет формировать и выводить на печать следующие виды отчетов:

1) подробно детализированный отчет о движении товаров, с возможностью детализации по отдельным типам товаров, организациям поставки, складам, актуальности товаров в требуемый отрезок даты;

2) отчет о количестве товаров во всех имеющихся складах;

3) складская документация (акты списания, накладные на отпуск материала со склада, логистические накладные, сводные отчеты о продажах, инвентаризационные ведомости, заявки на получение материалов со склада, поставщики номенклатуры...);

4) другие отчеты.

Ранее указанные формы отчетности позволяют быстро и эффективно управлять товарными потоками, вовремя вести учет остатков и формирование заказов на необходимый товар.

Список полезных функциональных частей для учета движения товаров на складе очень широк, и включает в себя такие позиции как:

1) ведение основных товарных операций:

- приход;
- расход;
- возврат;
- инвентаризация.

2) многоуровневая номенклатура;

3) товарный учет:

- распределенный по видам;
- распределенный по дополнительным подкатегориям и признакам;

4) актуальная информация:

- остатки в товарном эквиваленте;
- движения товарных остатков;
- хранение необходимых дополнительных характеристик о товаре;
- поставщики;
- другие;

5) формирование прайс-листов.

1.2. Обзор аналогов

Первым рассматриваемым вариантом будет программа «МойСклад» - это облачный сервис для управления торговлей, разработанный для малых и средних торговых компаний и индивидуальных предпринимателей. Сервис предназначен для опта, розницы и торговли во всемирной сети. «МойСклад»

поддерживает синхронизацию с интернет-магазином и рабочим местом кассира.

Основные возможности программы:

- управление любым количеством торговых точек и юр. лиц;
- полноценный складской учет;
- оформление закупок;
- печать всех первичных документов, ценников, товарных чеков;
- подключение рабочего места кассира (плюс сканер и ККМ);
- поддержка ЕГАИС;
- регистрация движения денежных средств;
- учет расчетов с контрагентами;
- работа с контрагентами (CRM);
- онлайн интеграция с интернет-магазином;
- удобные мобильные приложения

№	Пров.	Время	Контрагент	Организация	Сумма	Валюта	Выставлено с...	Оплачено	Отгружено	Зарезервиро...	Статус	Изменен	Изменил
70483	✓	17.10.2017 13:22	Новый клиент, источ...	ООО "РОМАШКА"	0,00	руб	0,00	0,00	0,00	0,00	Новый	17.10.2017 17:12	Иванов Д.
70482	✓	17.10.2017 13:21	Новый клиент, источ...	ООО "РОМАШКА"	0,00	руб	0,00	0,00	0,00	0,00	Новый	17.10.2017 17:12	Иванов Д.
70481	✓	17.10.2017 13:21	Новый клиент, источ...	ООО "РОМАШКА"	0,00	руб	0,00	0,00	0,00	0,00	Новый	17.10.2017 17:12	Иванов Д.
BB1	✓	22.06.2017 15:38	Иванов Петр	ООО "РОМАШКА"	5 387,10	руб	0,00	0,00	0,00	0,00	Отгружен	17.10.2017 17:11	Иванов Д.
BB2	✓	22.06.2017 15:38	Иванов Петр	ООО "РОМАШКА"	6 156,20	руб	0,00	0,00	0,00	0,00	Собран	17.10.2017 17:11	Иванов Д.
BB3	✓	22.06.2017 15:38	Иванов Петр	ООО "РОМАШКА"	2 302,20	руб	0,00	2 302,20	2 302,20	0,00	Собран	17.10.2017 17:11	Иванов Д.
BB4	✓	22.06.2017 15:38	Иванов Петр	ООО "РОМАШКА"	1 899,00	руб	0,00	0,00	0,00	0,00	Отгружен	17.10.2017 17:12	Иванов Д.
BB5	✓	22.06.2017 15:38	Иванов Петр	ООО "РОМАШКА"	1 999,00	руб	0,00	1 999,00	1 999,00	0,00	Подтвержден	17.10.2017 17:11	Иванов Д.
BB6	✓	06.02.2017 09:43	Иванов Петр	ООО "РОМАШКА"	1 500,00	руб	0,00	0,00	0,00	0,00	Возврат	17.10.2017 17:11	Иванов Д.
сск дебе...	✓	17.01.2017 10:47	Иванов	ООО "РОМАШКА"	957,51	руб	0,00	0,00	0,00	0,00	Отменен	17.10.2017 17:11	Иванов Д.
сск дебе...	✓	13.01.2017 14:03	Иванов	ООО "РОМАШКА"	50,00	руб	0,00	0,00	0,00	0,00	Доставлен	17.10.2017 17:11	Иванов Д.
BB7	✓	22.11.2016 18:17	ООО "Покупатель"	ООО "РОМАШКА"	2 250,00	руб	0,00	0,00	0,00	0,00	Доставлен	17.10.2017 17:10	Иванов Д.
00001	✓	15.04.2016 12:05	Розничный покупатель	ООО "РОМАШКА"	11 111,00	руб	0,00	0,00	0,00	0,00	Доставлен	17.10.2017 17:10	Иванов Д.
					33 612,01		0,00	4 301,20	4 301,20	0,00			

Рисунок 1.2.1- Интерфейс «МойСклад»

Данная программа имеет следующие минусы:

- отсутствие большей части отчетности (в т.ч. отчет продаж по категориям);
- неудобный интерфейс раздела закупок;
- медленное исправление ошибок ПО;
- высокая стоимость;
- проблемы синхронизации;
- медленная работа.

Также при необходимости использования подобных комплексов не стоит забывать и про программу «Склад+».

Складской учет "Склад+" - простая удобная программа, в которой есть все необходимое:

- создание документов по приходу и расходу ТМЦ;
- печать первичных документов;
- гибкое ценообразование с расчетами от закупочной цены;
- экспорт прайс-листа в таблицу Microsoft Excel;
- формирование предварительных заявок;
- группировка товаров по разделам;
- универсальная фильтрация данных по товарам, разделам, количеству.

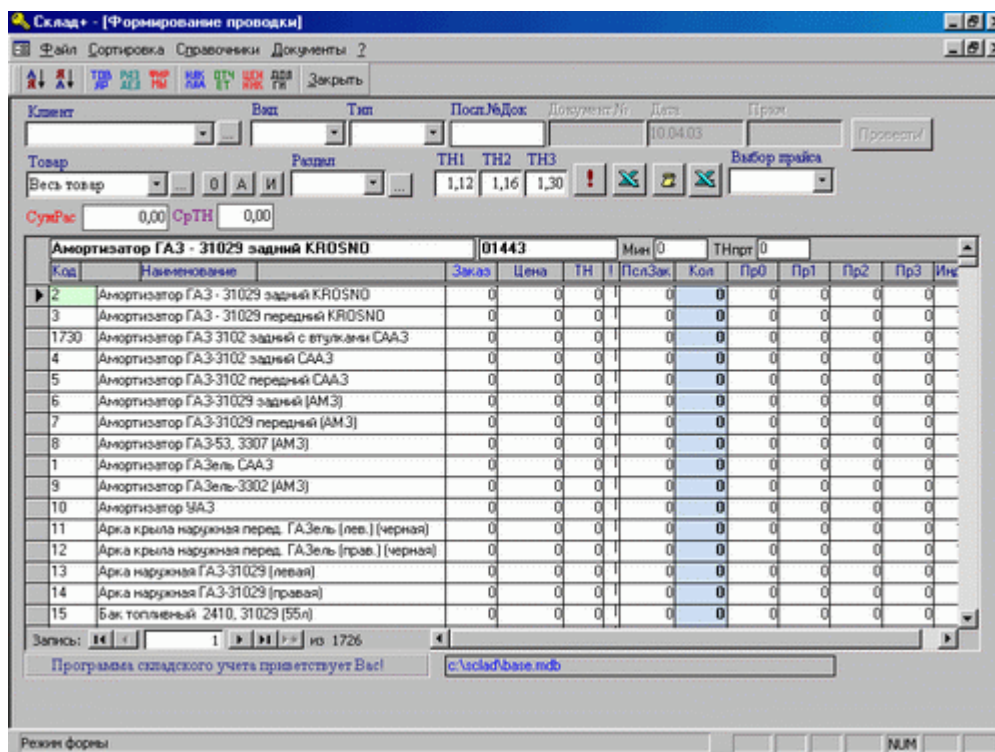


Рисунок 1.2.2 - Интерфейс «МойСклад»

Рассмотрим недостатки данного программного решения:

- ограниченные возможности в работе с документацией;
- устаревший интерфейс со сложным взаимодействием;
- отсутствие обновлений, слабая поддержка разработчиками;
- низкие мощности, вызывающие снижение скоростей работы;
- ограниченный функционал, отсутствие некоторой документации;
- отсутствие синхронизации со сторонним ПО;
- необоснованная стоимость;
- нестабильная работа.

Также рассмотрим готовые предложения от самой компании 1С, которые в теории могли бы подойти требуемому предприятию.

«1С Управление торговлей» – универсальный продукт от компании 1С, направленный на повышение эффективности бизнеса торгового предприятия. Набор опций, входящий в данный программный продукт внушительен, и изобилует различными функциями [1].

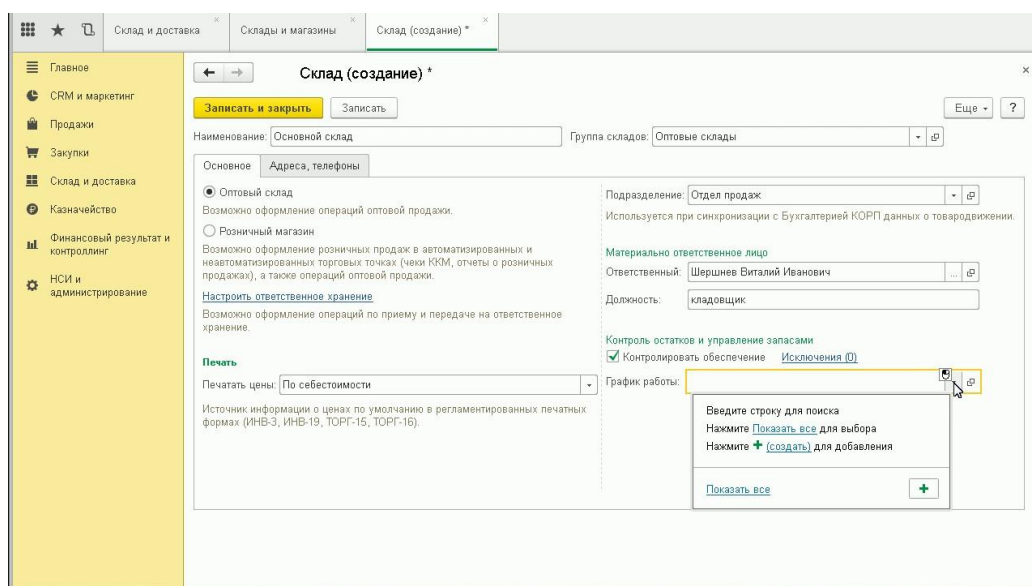


Рисунок 1.2.3 – Интерфейс «1С Управление торговлей»

Но в данном моменте находится как основной плюс, так и основной минус данного решения. Многим предприятиям внушительная часть опций попусту не нужна, и будет скорее мешать взаимодействовать с интерфейсом, нежели помогать с эффективностью работы. Также не стоит забывать о том, что программные продукты компании во многом направлены на свои сферы, и для полноценной работы предприятия потребуется докупать и другие программы, как например «1С Бухгалтерия» для ведения бухгалтерской отчетности. Данные манипуляции вызывают затраты, как времени, так и финансовых средств. К тому же программный продукт редко используется в чистом виде, для работы под конкретное предприятие требуется вносить свои коррективы. И тут кроется следующий недостаток, при изменении изначальной конфигурации программный продукт перестает поддерживать обновления. И со временем получить обновление купленной предприятием программы будет невозможно.

Представленные на рынке предложения не подходят в полной мере для решения поставленных задач, ввиду того, что не предоставляют стабильную работу, удовлетворяющую всем требованиям клиента. Итоговый пользователь должен иметь решение, способное улучшить работу предприятия и ускорить требуемые процессы. Но рассмотренные выше программные решения в данном вопросе не способны в полной мере удовлетворить требованиям, имея

слишком высокую стоимость, слабую поддержку разработчиками и отсутствие важной части функционала и документационных операций.

1.3. Постановка задач

Работники предприятия вынуждены выполнять огромное количество действий по поиску информации по предприятию в целом. Выполнение этой работы вручную требует значительного времени и терпения, ввиду большого количества действий, связанных с текстовой информацией.

Цель работы – разработка комплекса для учета движения товаров на складе предприятия.

Система будет оптимизирована для использования работниками предприятия такими как: отделы закупа и сбыта, заведующий складом, занимающиеся учетом товаров на складах, и внесением в расчет вновь поступивших товаров. В то же время программа должна осуществлять ведение статистики по сотрудникам. Графики работы, расчет премий и зарплат, исходя из пропущенных рабочих дней и продаж по каждому работнику, как уточнило предприятие, очень важны для их работы. Также важными для отдела менеджмента фирмы оказались диаграммы ERM (диаграмма Ганта), который требуется добавить в финальную версию продукта. Финальный продукт должен обладать требуемыми предприятием функциями, не перегружая физические носители информации предприятия, имея в то же время понятный интерфейс, без излишних функций.

Разрабатываемая информационная система позволит кардинально снизить затрачиваемое время работы сотрудников и упростит процесс получения различной сводной информации.

Итак, рассмотрим задачи решаемые данным ПО:

- обеспечение хранения и обработки данных и последующий вывод их на печать;
- минимизировать трудовые затраты, затрачиваемые на принятие информации и её обработку;
- наличие удобных форм формата «ввод-вывод» с легким усвоением управлением;
- открытый формат уведомления клиентов о имеющихся проблемах или ошибках;
- вывод требуемых форм документов на печать, а именно: накладная заказа клиента, счет на оплату, лист инвентаризации;
- возможность создания документации и отчетов;
- ведение зарплатных ведомостей и графиков.

2. Проектирование программного продукта

В данной части будет рассмотрен процесс проектирования базы данных. Раскроем такие понятия, как: строение и функциональная структура программного обеспечения и приложений. Крайне важно описать причины почему именно этот тип базы данных был выбран и как принятое решение будет влиять на дальнейшее построение приложения. Исходя из этого, необходимо учесть все преимущества и недостатки выбранного типа базы данных.

Рассмотрим предметную область, что будет описана в нашем проекте: необходимый перечень сущностей и табличных форм базы данных, для полноценного функционирования

БД должна иметь четко-выстроенные, прошедшие сортировку таблицы, со связанной между собой и распределенной по необходимым категориям информацией.

2.1 Структура программного обеспечения

При создании любого приложения используются различные программные инструменты и основы, являющиеся частями так называемой структуры ПО.

Используемое программное обеспечение, назовем его инструментами, распределено по некоторым структурным группам, а именно:

- системное: предназначено для широкого круга действий общего использования. Причислим к ним ОС с их подвидами и разновидностями, сопутствующие программные утилиты, системы технического типа действия, для решения проблем в работе механизма;

- вспомогательное: программное обеспечение, предназначенное для непосредственного взаимодействия с персоной и выполнения её прикладных задач. К ним могут быть причислены: различные видео-, графические и текстовые редакторы, компьютерные игры и т.д.;

- инструментальное ПО: комплекс или совокупность программ, которые используют для создания программ ПО, в том числе прикладного и системного. Это системы, позволяющие разработчику взаимодействовать с кодом того, или иного приложения, а именно создавать, корректировать и производить запуски. Актуально на данный момент будет назвать подобные системы интегрированными средами разработки, исходя из наличия в них огромного числа инструментов взаимодействия с языками программирования и систем управления БД.

Не стоит забывать и о базовой системе ввода-вывода (Input-Output) именуемой BIOS (Basic Input-Output System). Данная система находится на

более начальном уровне, а именно на уровне аппаратного обеспечения компьютера.

Создание приложения проводилось на операционной системе Windows 10 актуальной версии, с типичной для UNIX-систем графической оболочкой. Место данной операционной системы в процессе разработки нашего приложения вы можете видеть на схеме 2.1.1.

Нашим главным помощником в оформлении и описании проекта был пакет программ американской корпорации Microsoft, именуемый MS Office, которому в процессе создания различных диаграмм таблиц помогала web-программа «drawio».

Разработка же основного программного обеспечения проводилась в технологичной среде 1С: Предприятие.

Ниже представлена блок-схема построения приложения:



Рисунок 2.1.1 – Структура ПО

Инструментальное программное обеспечение несомненно очень важная составляющая разработки, позволяющая облегчить, ускорить и создавать ПО более наглядно.

Неотъемлемой частью большей части приложений, работающих с информацией, является наличие базы данных, такого структурированного хранилища для информации.

База данных безусловно должна иметь доступ к определенному количеству памяти устройства для оптимального функционирования и взаимодействия с большим количеством информации.

Проектирование базы данных – трудоемкий процесс, требующий от разработчика немалое количество потраченного времени и качественное использование логических и умственных способностей.

Логика структуры приложения несомненно важна, так как от этого будет зависеть насколько удобно проходит взаимодействие разработчика с программой после переноса логики в таблицы БД.

Согласно задачам комплекса учета, имеется острая необходимость в создании собственной базы данных, поскольку работникам организации требуется место для хранения всей информации, связанной с деятельностью складского комплекса, начиная количеством и типами имеющихся товаров, заканчивая документацией организации.

В нашей базе данных будет информация о контрактах, поставщиках, действиях персонала, имеющихся товарах, их состоянии и т.д. в определенный период времени.

Важно построить базу данных с логически продуманной структурой, без лишних связей и лишних компонентов, не утяжеляя тем самым взаимодействие пользователя с программой.

Тем самым ускорится и значительно упростится решение задач и запросов как для персонала, взаимодействующего с программой, так и для аппаратного оснащения предприятия. В качестве примера можно привести библиотеку с логически продуманной сортировкой книг и их удобным расположением в помещении. Так же и в нашей базе данных пользователю должно быть удобно найти определенный сегмент информационной базы, без лишних действий и хаотичного потока ненужной информации.

Также несомненно важным моментом будет отсеивание лишней информации, что будет излишне бессмысленно нагружать аппаратную основу. Необходимо использовать базу данных, как хранилище для тех значений, что непосредственно будут необходимы для взаимодействия с приложением и успешного функционирования комплекса. Часто отсутствие подобной ответственной работы над список значений не позволяет конечным пользователям почувствовать всю скорость и эффективность нынешних СУБД. Актуальные на данный момент версии СУБД имеют ряд отличных возможностей для усовершенствования программных возможностей, в особенности возможность изменения полей в требуемых таблицах, данное действие помогает в дальнейшем контролировать количество табличных структур.

Нам всем известно, что базы данных в современности разделены на две большие группы, а именно: SQL и NoSQL, реляционные и нереляционные базы данных соответственно. Отличия данных систем друг от друга заложены еще в самой основе базы данных, разными типами поддерживаемых данных, их проектировкой и взаимодействием с данными.

SQL базы данных являются хранилищем для описаний конкретных объектов реального мира. Это может быть, как содержимое мешка с овощами или данные студентов отсортированные по типам в таблицах, чья структура принята еще на фазе проектирования хранилища.

Персона	Друзья
Лиза	{Саша, Маша}
Саша	{Лиза, Игорь, Влад}
Маша	{Лиза, Влад}
Игорь	{Саша}
Влад	{Саша, Маша}

Рисунок 2.1.2 - Пример реляционного типа. Структурированная информация

NoSQL базы данных же функционируют в другом ключе. В случае, если в реляционных базах данных информация будет разбита по некоторым признакам в таблицы, в нереляционной же базе эти же данные будут храниться без разбиения в целостной независимой сущности.

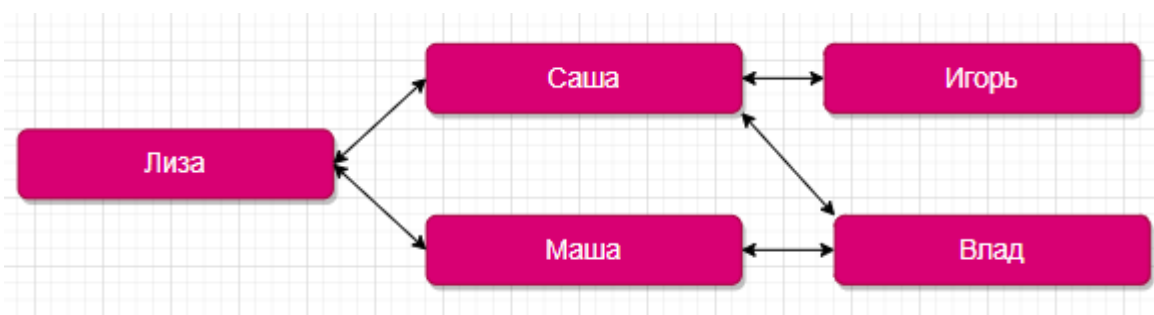


Рисунок 2.1.3 - Пример нереляционного типа. Безсхемная структура

Каждый вид базы данных из представленных выше имеет свою нишу для использования, исходя из особенностей конфигурации и зоны воздействия базы данных в конкретном проекте. К примеру, NoSQL базы данных открыты к масштабированию, в тот же момент как SQL базы данных больше похожи на готовое хранилище для уже структурированных по каким-либо параметрам данных.

Итак, давайте же разберемся к каким проектам подойдут реляционные базы:

- 1) Присутствуют логические требования к информации.
- 2) Крайне важна целостность информации.
- 3) Вносимая информация может получить структуру по каким-либо параметрам.

И отличительные качества проектов, которым будут удобны нереляционные базы:

- 1) Отсутствие четко прописанных и определенных рамок для данных.
- 2) Цели проекта несколько размыты и допускают их дальнейшее изменение или доработку, при этом важна возможность немедленного начала разработки.
- 3) Требуется скорость обработки информации
- 4) Масштабируемость.

Но в итоге всё-таки не стоит рассматривать данные виды баз данных категорично как два совершенно обособленных друг от друга инструмента. Всё чаще сейчас в крупных проектах используются оба вида баз данных, с методично подобранной локальной зоной действия.

Для комплекса учета склада, реляционная база данных будет подходящим решением, ввиду четко поставленных задач и отсутствия абстрактных требований, также вся информация, используемая элементами программы может быть представлена в виде пары взаимосвязанных таблиц. В нашем случае база данных не потребует дальнейших изменений в последние этапы разработки, поэтому итоговую структуры БД составим на начальном этапе проектирования. Обращаться с потоками информации придется с четкими алгоритмами, без размытости, ввиду того, что процесс обработки информации должен быть прописан в начале, и не потребует дальнейших изменений.

В данной части проекта необходимо рассмотреть и проанализировать предметную область нашей работы для дальнейшего корректного проектирования структуры БД.

Анализ предметной области несомненно одно из важнейших процессуальных фаз в создании приложения, важность заключается в том, что данный этап позволяет рассмотреть данные, требующиеся в вопросе корректного использования приложения пользователем и хранения информации в базе данных.

Пошаговая работа в рассмотрении данного вопроса очень важна, так как необходимо проверить каждый из этапов, от начального функционала приложения, до конечного взаимодействия с пользователем. Тем самым будет получена необходимая информация о данных, что будут востребованы пользователем и какие действия пользователь будет с ними осуществлять. Проведя данный анализ, нужно добавить необходимую информацию и функционал в структуру приложения.

Итак, с чего же начинается процесс использования приложения пользователем: вход в систему.

В целях безопасности и в целом осуществления входа в приложение нашему пользователю необходимы уникальные поля, а именно логин или идентификатор, и пароль, удовлетворяющий стандартам безопасности.

Логин может представлять из себя различную информацию: никнейм, телефонный номер, персональные данные и т.д. В нашем случае используется телефонный номер ввиду его уникальности и упрощении тем самым процесса запоминания данных пользователем, и облегчении распознавания данного пользователя системными работниками.

Пароль – символьный набор, подобранный пользователем для защиты собственной учетной записи. Важное уточнение в описании того, что из себя представляет пароль, это его хранение в зашифрованном виде и отсутствие доступного права просмотра к нему у системного администратора.

При отсутствии регистрации у пользователя и отсутствии учетной записи соответственно, пользователь переходит к окну регистрации. Там пользователь должен будет указать данные, которые будут использованы для создания учетной записи, а именно: номер телефона, в качестве идентификатора и символьный набор (пароль). В данном моменте имеется возможность добавить ФИО пользователя для большей персонализации записи, все уточняющие характеристики, в том числе должность данного сотрудника.

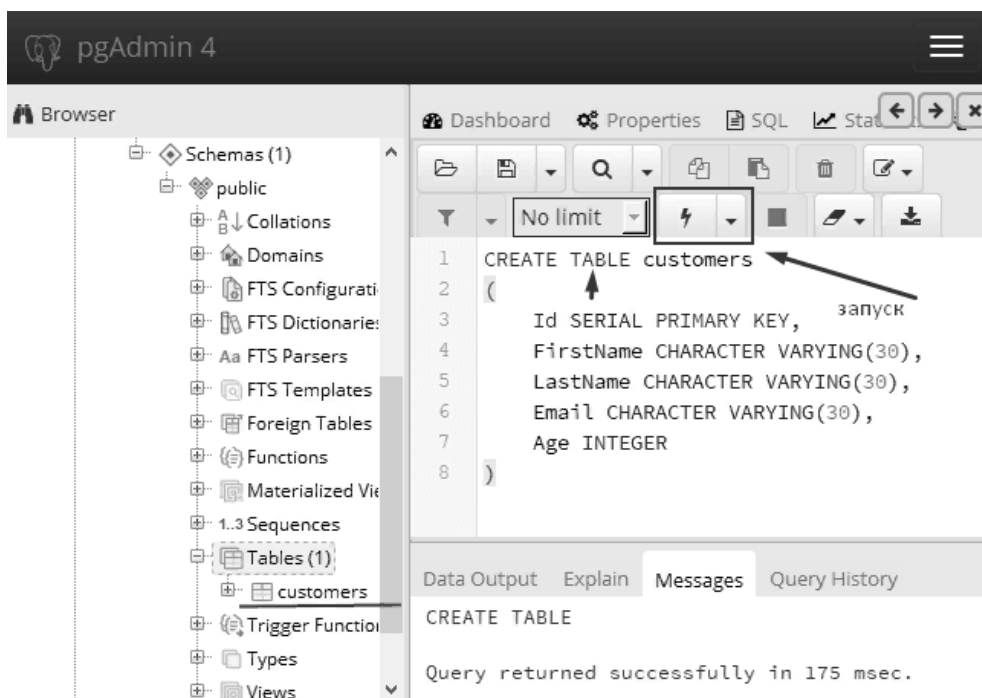


Рисунок 2.1.4 – Запрос для создания таблицы

На данном этапе будет проведено создание таблицы «Пользователи». Таблица призвана для хранения полей пользователя, персонализирующих его: телефонный номер, ФИО, рабочая должность, пароль.

После идентификации и прохождения процедуры входа в систему, пользователь имеет выбор, что использовать дальше: планировщик или диалоговое окно.

Сообщения должны иметь поля, по которым их можно будет в дальнейшем хранить и сортировать, соответственно для успешной и корректной работы и хранения диалоговых сообщений, должны иметь такие поля, как: автор сообщения, конечный получатель, содержимое сообщения, индивидуальный идентификатор, и поле описывающее момент времени отправки.

Поле-хранилище идентификатора требуется для того, чтобы конечный получатель имел информацию, повествующую ему, от кого пришло данное сообщение, к какому чату относится исходя из данных полей.

Следующим полем будет рассмотрено поле-хранилище для идентификатора конечного пользователя получателя, а именно персона, что должна получить сообщение, служит для того, чтобы указать конкретный чат для отправки, а не отправлять сообщение произвольно подобранному пользователю, абсолютно никак не связанным с данным чатом.

Также имеется поле с содержимым сообщения, а именно той информацией, что пользователь-отправитель хочет предоставить пользователю-получателю.

Для маркировки и персонализации сообщения используется поле-идентификатор. Данное поле делает сообщение уникальным и отличным от остальных полей, даже если два сообщения будут схожи по всем полям, то по данному полю всегда будут отличаться, соответственно это позволит администратору с легкостью находить необходимое сообщение. Тем самым администратор сможет корректировать, удалять, и совершать требуемые действия с сообщением в дальнейшей работе приложения, что несомненно важно для развития приложения и его корректной работы.

Важным полем для описания сообщения является время отправки, время отправки характеризует и дает большую точность смыслу сообщения, к примеру, если получатель прочел сообщение лишь по прошествии пары дней после отправки и сообщение уже потеряло некоторую актуальность, то при прочтении даты отправки получатель может верно понять содержимое данного текста.

Дата отправки несомненно важный параметр сообщения, позволяющий чату быть структурированным и понятным.

Дальше будет создана вторая таблица базы данных «Сообщения» наполненную данными сообщениями и названную так соответственно своему назначению. Данная таблица будет хранить в себе все поля, относящиеся к сообщению и описывающие его. В том числе информацию о получателе, отправителе, дата, идентификатор и текст сообщения.

+ Параметры				msg_id	msg_time	msg_text
<input type="checkbox"/>				8	2016-05-16 07:29:46	Привет всем кто в этом чате!
<input type="checkbox"/>				9	2016-05-16 07:29:52	Салют, как дела?
<input type="checkbox"/>				10	2016-05-16 07:29:59	Та так, а у тебя как?
<input type="checkbox"/>				11	2016-05-16 07:30:07	Та норм в принципе

Рисунок 2.1.5 – Пример того, что действительно содержится в БД

Результатом данного запроса для создания таблицы базы данных будет:

Messages	
msg_id:	integer
+ msg_time:	datetime
+ msg_sentby:	binary
+ msg_sentto:	binary
+ msg_text:	string

Рисунок 2.1.6 – Данные таблицы

Следующим звеном в создании будет таблица, хранящая данные о процессах в планировщике.

Каждая задача, добавляемая в хранилище базы данных, имеет своего пользователя, который создал эту задачу. Таким образом, можно выбирать только те задачи из базы, которые принадлежат конкретному пользователю и отображать только его задачи. Это поможет избежать появления в списке задач пользователя тех задач, которые ему не принадлежат.

Далее, каждая задача представляет собой какое-либо слово, набор слов или предложение. Это является самым основным полем в таблице, которое несет в себе главную информацию во всем сервисе.

Далее идет дата выполнения задачи. В это поле базы данных записывается та дата, на которую записывается задача. Это поле позволяет пользователю из всего списка задач, выбрать те задачи, которые относятся к определенной дате. Таким образом, заходя в планировщик, пользователь

первым делом видит все те задачи, которые предназначены для выполнения в текущую дату.

Каждая задача имеет статус прогресса. Статус прогресса задачи показывает на каком этапе сейчас находится выполнение задачи. Пользователь мог не приступать к выполнению задачи, что будет соответствовать статусу «не выполнялась», либо мог уже выполнить задачу, значит, статус в данном случае будет иметь значение «выполнена». Также статус имеет два промежуточных состояния: задача может находиться «в процессе» выполнения, либо может быть «отложена», если пользователю надобилось переключиться на другую задачу, либо есть нужда ждать какого-либо действия другого сотрудника, чтобы продолжить работу.

В итоге будет создана третья таблица базы данных с названием «Задачи». Эта таблица будет хранить в себе все данные относительно каждой задачи в полях, названия которых были перечислены выше: пользователь, текст задачи, статус прогресса задачи, дата выполнения задачи.

Результат запроса на все данные из таблицы, которая уже заполнена тестовыми значениями выглядит следующим образом:

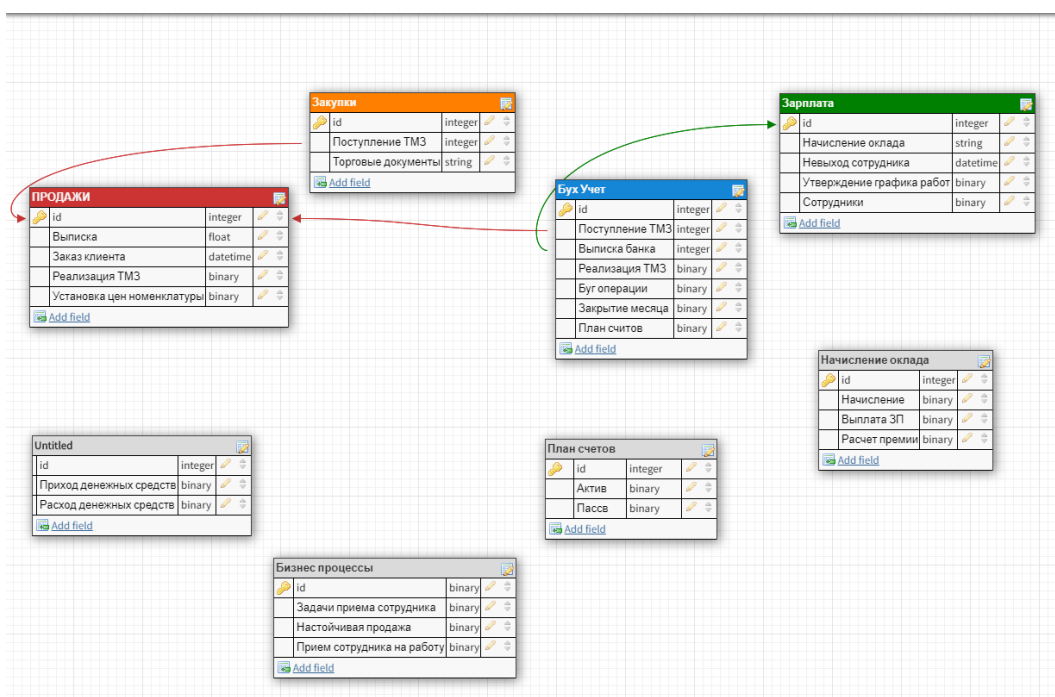


Рисунок 2.1.7 – Данные таблицы

Далее следует то, как устроено позиционирование таблиц в базах данных и то, какими связями объединены.

Показ таких таблиц позволяет увидеть какого типа информация должна будет в дальнейшем заполнять базу данных, и то в какой структуре будут находиться таблицы. Фигуры на схеме являются визуальным представлением

таблицы с входящими в нее полями. Важнейшие поля выделены специальными обозначениями.

По данной схеме можно увидеть связи таблиц и их взаимодействие по определенным полям.

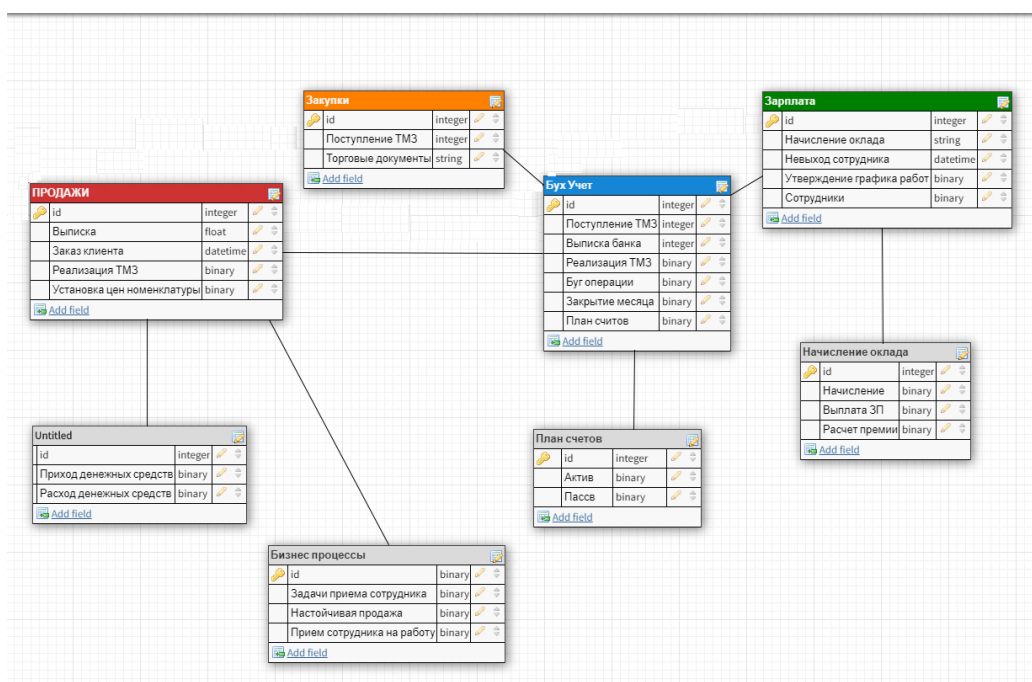


Рисунок 2.1.8 – Структура базы данных

Дополнительные инструменты в работе с базой данных крайне необходимы для точечной и более комплексной работы с информацией, в работе с базой данных использовался широко известный плагин *Suppose* с расширением *Database* в *IntelliJ Idea*. Соответственно используется для работы с базами данных в среде разработки, только на этот раз это производится напрямую. Тем самым уменьшается список приложений для работы, используя один лишь плагин, при помощи которого ведутся процессы разработки, отладку продукта применяя только функционал одного приложения.

Универсальность данного решения бесспорна ввиду лояльной системы интеграции и синхронизации. Все необходимые утилиты и драйверы программа сама предлагает установить на ваш компьютер, тем самым автоматизируя и сильно облегчая процесс поиска сопутствующего ПО в процессе разработки для программиста.

Ввиду построения всей данной системы на языке из списка высокоуровневых, а именно на *Java*, не обойтись без *framework-a*.

Первым на ум придет фреймворк *Hibernate*, в качестве распространенного и устоявшегося решения данного вопроса, но в нашем

случае была предпочтена альтернатива в лице MyBatis, имеющему немалое количество преимуществ. Методика работы MyBatis представляет из себя связывание алгоритмов Mapper интерфейсов с реляционными запросами.

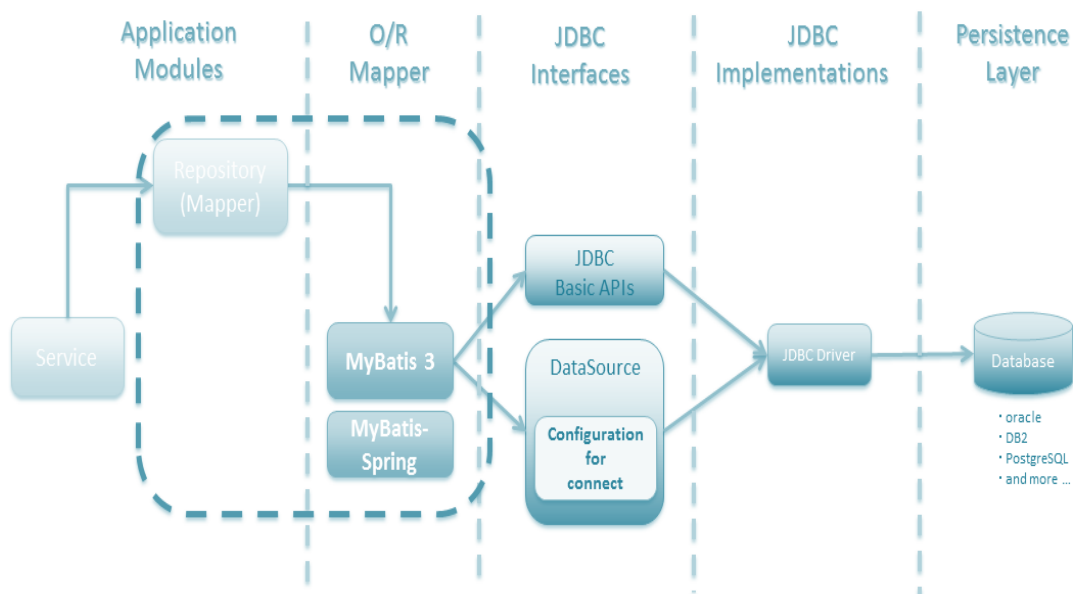


Рисунок 2.1.9 - Схема алгоритма включения фреймворка в схему работы программного комплекса

В его функционал не входит создание схематичных данных и сопутствующих действий. MyBatis скорее предоставляет возможность для трансформирования метод-вызова в конкретно существующий запрос в базу данных, и в последствии дает результат в виде возврата обычного метод-вызова, при использовании стороннего спец-приложения.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="test">
  <select id="selectData" parameterType="java.util.Map"
    resultType="String">
    SELECT COL2 FROM #{TABLE_NAME} WHERE COL1 = #{COL1}
  </select>

  <select id="selectData2" parameterType="java.util.Map"
    resultType="String">
    SELECT COL2 FROM ${TABLE_NAME} WHERE COL1 = #{COL1}
  </select>
</mapper>
```

Рисунок 2.1.10 - Пример включения плагина в код

На данном этапе важно построить функциональную структуры будущего приложения, послужит является отличным методом презентации возможностей и набора действий, что программа в силах сделать. Рассматривая подобную структурированную форму, сразу можно понять стоит ли данное приложение внимания и действительно ли решит требуемые задачи.

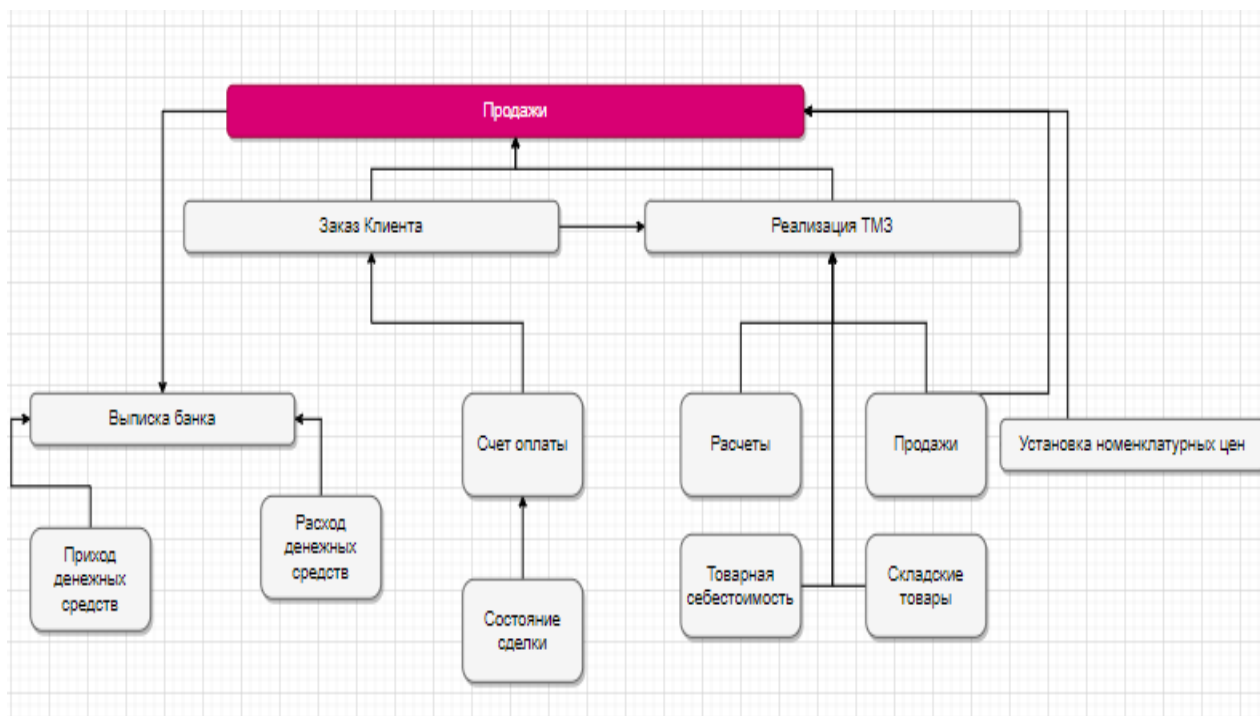


Рисунок 2.1.11 – Функциональная структура

В процессе проектировки комплекса учета движения товаров на складе предприятия Ногеса используется средство для визуального моделирования бизнес-процессов BPwin 4.0, что представляет собой действенный инструмент, помогающий в создании моделей, позволяющих проводить анализ, вести документацию и планы корректировки требуемых бизнес-планов. BPwin позиционируется как средство сбора информации о функционировании компании и визуального изображения данных в виде графической модели.

BPwin включает методы трех направлений: IDEF0, DFD и IDEF3, представляющих собой три точки зрения для анализа бизнес-процесса:

IDEF0 (Integration Definition for Function Modeling) – со стороны функциональности системы. Рассматривает бизнес-процесс в качестве набора работ, находящихся во взаимодействии друг с другом, в дополнение к этому рассматривает затраты ресурсов (производственных, людских, и в виде данных), что потребуется для конкретной работы.

BAW (Data Flow Diagramming) – со стороны потоков данных. Рассматривает потоки данных, осуществляя возможность наблюдения обмена данными во взаимодействии бизнес-функций внутри системы, но само взаимодействие остается без внимания данных диаграмм.

IDEF3 – со стороны последовательности работ. Дает более детальную картину, указывая на последовательность осуществления работ. Детальную картину не только основного развития процесса, но и логически продуманных альтернатив.

BPwin совершает качественную проверку выпускаемым моделям со стороны синтаксиса избранной методологии, связей между диаграммами и их ссылок, вкуче с широким комплексом проверок, для результата в виде качественной модели, а не в виде графического наброска. В то же время не отключая главных плюсов рисунка – легкость в создании и наглядный показ.

В контекстной диаграмме, изображенной ниже (рисунок 1), данные управления находятся в верхнем блоке (указания работникам, государственные стандарты), а информация на вход (Товарные данные, Клиентские данные, Информация о заказе товара, данные поставщика), последующая обработке, расположена в левой части блока, а выходной результат показан в правой части блока (Прайс-лист, накладная на расход, отчетная документация). Механизм (Складской персонал, подразделение сбыта, персонал поставок), совершающим операции, входит в нижнюю часть блока.

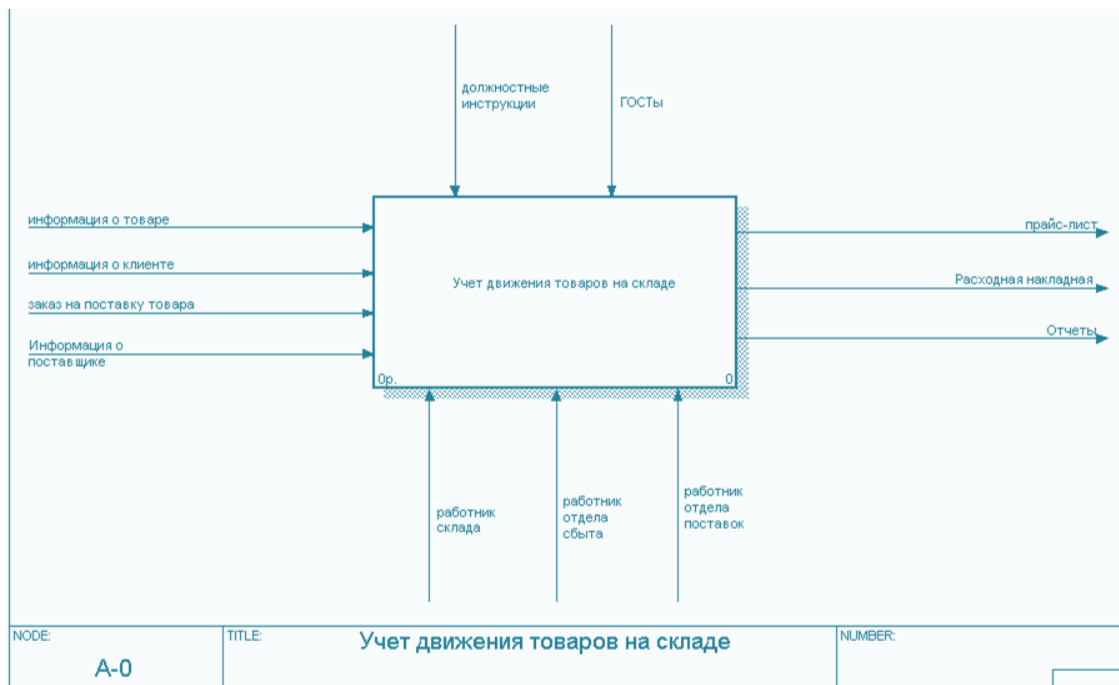


Рисунок 2.1.12 - Контекстная диаграмма

Последующий блок «Учет движения товаров на складе» преобразуется в 4 отдельные действия, представленных на декомпозиции процесса:

- прибытие товара в складском помещении;
- последующее хранение;
- выдача товара;
- создание и внесение заявки.

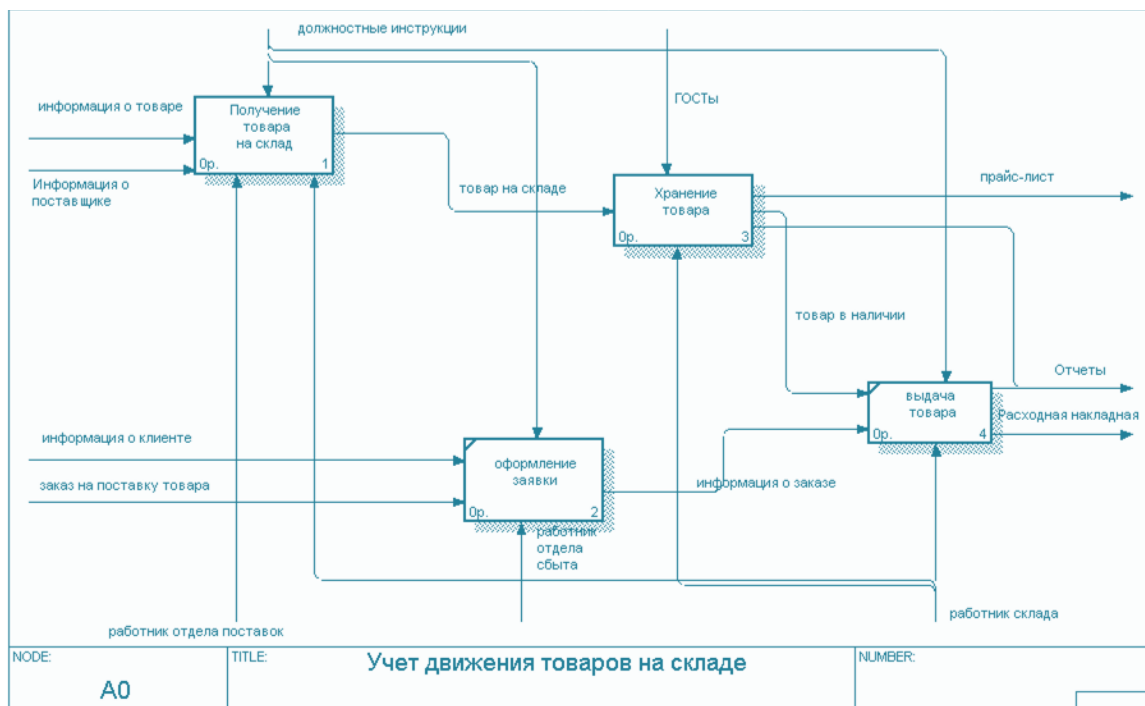


Рисунок 2.1.13 - Диаграмма декомпозиции

Блок «Прибытие товара на склад» имеет разбиение на 3 движения:

- внесение информации о поставщике;
- проверка информации;
- внесение товарной информации.

Диаграмма декомпозиции данного процесса представлена ниже.

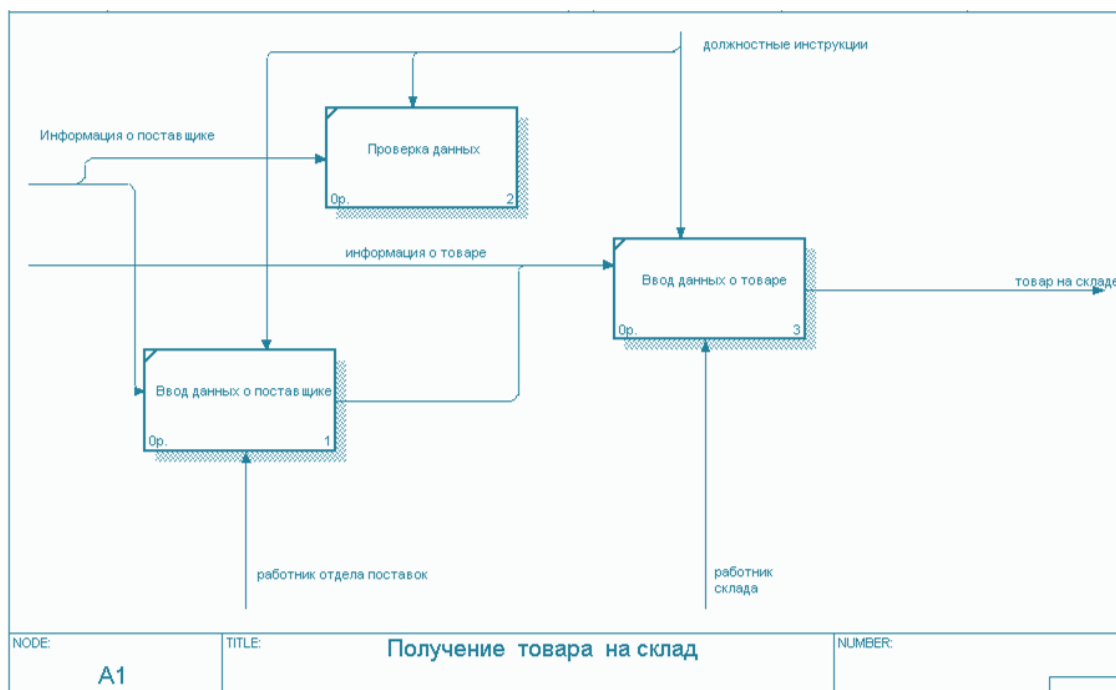


Рисунок 2.1.14 - Диаграмма декомпозиции

Создание информационной схемы осуществляется в несколько этапов:

1) Получение данных из разговора с заказчиком, изучение полученных данных, последующее выделение необходимой сущности (объекты предметной области, данные о которых требуют хранения). Все сущности имеют уникальный идентификатор и свойства:

- иметь один или несколько атрибутов, принадлежащие ей, либо наследуемые посредством связей;
- обладание ключами, одним или несколькими, являющимися идентификаторами каждого экземпляра;
- обладание любым числом связей с остальными сущностями.

2) Моделирование связей.

Связь – это бинарная ассоциация между двумя сущностями, показывающая соотношение и взаимодействие сущностей между собой в рассматриваемой предметной области. Распространены ассоциации родительской сущности с любым числом экземпляров сущностей-потомков.

Наименование связей должны быть уникальными, ввиду их идентификации посредством данных наименований, однако повторение в рамках модели допустимо. Также присутствуют степень и обязательность для связей. Направление связи отходит от родительской сущности

Имеются следующие типы связей:

- 1:1 (один к одному) – для верхних ступеней иерархической модели данных;

- 1:M (один ко многим) – связывание одного экземпляра от одной сущности с несколькими экземплярами другой сущности;
- M:N (многие ко многим) – использование в первичной фазе разработки диаграммы.

3) Определение атрибутов сущности.

Атрибут – именованная характеристика сущности, имеющая значение в рассматриваемой предметной области и используемой с целью классификации, идентификации и описания состояния конкретной сущности. Атрибут может быть описан или идентифицирован, при определении связи между сущностями идентифицирующие атрибуты, наследуемые в направлении от родительской сущности к потомственной сущности. Совокупность атрибутов или отдельный атрибут могут быть использованы для уникальной идентификации любого экземпляра сущности, именуемого первичным ключом. Атрибут, являющийся первичным ключом, обязательно располагается в верхнем отделении списка. Ключ не должен иметь значение 0 в любой из своих частей, должен быть заполненным и присутствовать. Внешний ключ (FK) – ключевой атрибут дочерней сущности, передающийся при связи сущностей.

Ниже будет показана схема базы данных, показывающей наличие в рассматриваемой базе данных семи таблиц, к соответствующим семи сущностям: Товары, товарные группы, расход товара, склад, поставщики, клиенты, приход товара. Рисунок также обладает атрибутами сущностей и их описанием. Связь сущностей в модели представлена связью один ко многим (1:M).

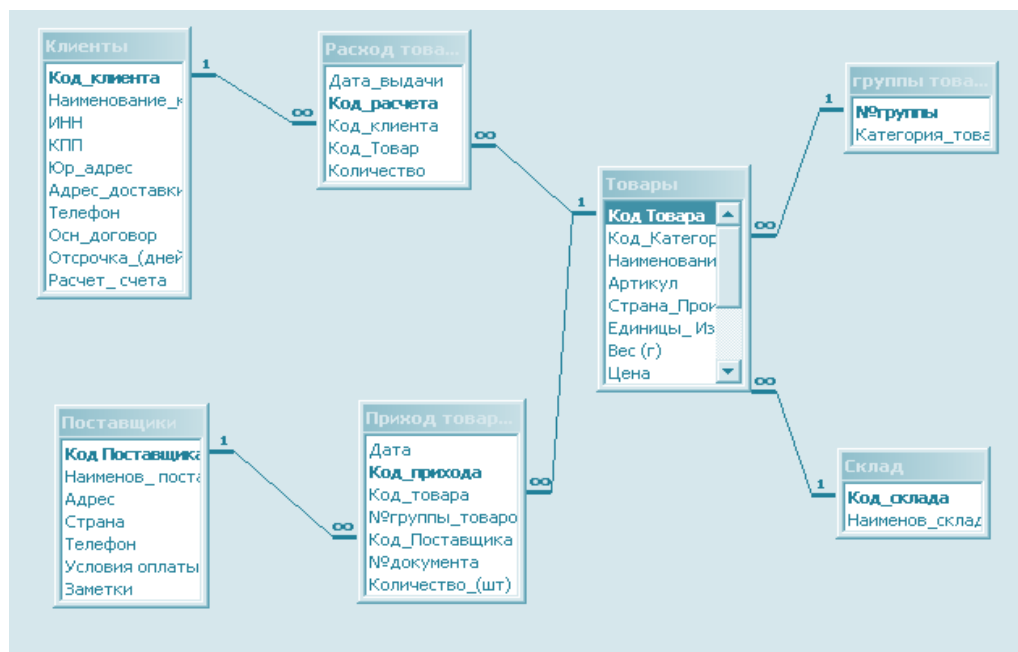


Рисунок 2.1.15 - Схема базы данных

Сущностям «Клиенты» и «Товары» необходимо обладать связями между собой, связь получается М:М, ввиду того, что количество клиентов больше одного может приобрести один товар, и к тому же несколько наименований товара могут быть проданы в одни руки, однако связи М:М в БД к использованию не рекомендованы, для этого вводится промежуточная сущность «Расход товара», связывающаяся с сущностью «Клиенты» по ключевому полю «Код_клиента» используя связь 1:М. В то время как сущность «Товары» связана с сущностью «Расход товара» в ключевом поле «Код_товара» связью 1:М.

Поставщики осуществляют поставку товара в складское помещение. Один представитель-поставщик осуществляет доставку как одного, так и нескольких товаров, а также несколько поставщиков могут быть поставщиками одного товара. Следовательно, вновь имеем связь М:М, в этом моменте вводится промежуточную сущность «Приход товара» для избавления от данного типа связи, промежуточная сущность подсоединяется к сущности «Поставщики» и сущности «Товары» связью 1:М.

Связью один-ко-многим связаны такие сущности, как «Группы_товаров» и «Товары», ввиду того, что несколько товаров могут принадлежать к одной и той же группе.

Ввиду того, что на одном складском помещении могут находиться несколько наименований товара, следовательно сущности «Склад» и «Товар» соединяются связью типа 1:М.

Схема полученная таким образом будет основой нашей базы данных для проектируемой системы.

Создадим базу данных, в качестве физического представления информации, содержащую общий файл расширения .mdb. В нижеприведенной таблице будут указаны поля таблицы, указав их идентификаторов, типы данных и т.д.

Таблица 1. Структура таблицы Товары.

Имя поля	Тип поля	Размер поля	Ограничения целостности
Код товара	Счетчик	Длинное целое	Primary Key
Код категории	Числовой	Длинное целое	Not Null
Название товара	Текстовый	50	Not Null
Артикул	Числовой	Длинное целое	Not Null
Страна_Происхождения	Текстовый	50	Not Null
Единицы Измерения	Текстовый	10	Not Null
Вес (г)	Числовой	Длинное целое	Not Null
Цена	Денежный	Авто	Not Null

В_наличии	Логический	Авто	Null
Сертификат	Текстовый	50	Not Null
Срок_действия_до	Дата/время	Краткий формат даты	Not Null
№ склада	Числовой	Длинное целое	Not Null
Срок_годности_товара	Дата/время	Краткий формат даты	Not Null
Количество_товара	Числовой	Длинное целое	Not Null

Таблица 2. Структура таблицы Клиенты

Имя поля	Тип поля	Размер поля	Ограничения целостности
Код клиента	Счетчик	Длинное целое	Primary Key
Имя клиента	Текстовый	50	Not Null
ИНН	Числовой	Действительное	Not Null
КПП	Числовой	Действительное	Not Null
Юр_адрес	Текстовый	50	Not Null
Адрес_доставки	Текстовый	50	Not Null
Телефон	Текстовый	50	Null
Основной_договор	Текстовый	50	Not Null
Отсрочка_(дней)	Числовой	Длинное целое	Not Null
Расчет_счет	Текстовый	50	Null

Таблица 3. Структура таблицы Поставщики

Имя поля	Тип поля	Размер поля	Ограничения целостности
Код_поставщика	Счетчик	Длинное целое	Primary Key
Данные_поставщика	Текстовый	50	Not Null
Адрес	Текстовый	50	Not Null
Страна	Текстовый	50	Not Null
Телефон	Текстовый	50	Null
Условия_оплаты	Текстовый	50	Not Null
Заметки	Текстовый	50	Null

Таблица 4. Структура таблицы Группы товаров

Имя поля	Тип поля	Размер поля	Ограничения целостности
№ группы	Счетчик	Длинное целое	Primary Key
Категория_товара	Текстовый	50	Not Null

Таблица 5. Структура таблицы Склад

Имя поля	Тип поля	Размер поля	Ограничения целостности
Код_склада	Счетчик	Длинное целое	Primary Key
Наименование_склада	Текстовый	50	Not Null

Таблица 6. Структура таблицы Приход товаров

Имя поля	Тип поля	Размер поля	Ограничения целостности
Дата	Дата/время	Краткий формат даты	Not Null
Код_прихода	Счетчик	Длинное целое	Primary Key
Код_товара	Числовой	Длинное целое	Not Null
№ группы_товаров	Числовой	Длинное целое	Not Null
Код_поставщика	Числовой	Длинное целое	Not Null
№ документа	Числовой	Длинное целое	Not Null
Количество_(шт)	Числовой	Длинное целое	Not Null

Таблица 7. Структура таблицы Расход товаров

Имя поля	Тип поля	Размер поля	Ограничения целостности
Дата_выдачи	Дата/время	Краткий формат даты	Not Null
Код_расчета	Счетчик	Длинное целое	Primary Key
Код_клиента	Числовой	Длинное целое	Not Null
Код_товара	Числовой	Длинное целое	Not Null
Количество	Числовой	Длинное целое	Not Null

Таблицы содержат в себе информацию о приходе и расходе товаров. В случае изменения каких-либо данных в таблицах, целостность базы не нарушается, происходит замена предыдущей информации, на обновленную, включая запросы, где данные поля были использованы. К тому же для удобства пользования данные добавляемые в таблицы могут быть выбраны из подобранных программой, для уменьшения выбора информации заново.

Следующим шагом будет процесс создания структуры меню – а именно процесс разработки и проектирования структуры ИС. Ниже представлена структура информационной системы.



Модуль работы «Справочники» построен из 3 последующих справочников, именуемых как:

- «Клиенты»;
- «Поставщики»;
- «Товары».

Рабочие цели данных справочников включают в себя: поиск и просмотр данных по товарам и покупателям, поставщикам. К тому же можем отнести создание и обработку справочной информации.

В качестве базы данных приведенного модуля приведены следующие таблицы:

- «Клиенты»;
- «Товары»;
- «Поставщики».

Структурная реализация показана в таблицах 1-3. Информация в информационную базу через специальные формы, вызывающиеся из основного меню программы.

Модуль, предназначенный для выполнения операций, над имеющейся информационной базой состоит из 3 операций:

- приход товара;
- расход товара;
- прайс-лист.

Операция «Приход товара» выполняет функцию вноса информации о товаре и слежения за документами прихода. Удобный интерфейс облегчает поиск и выбор требуемого поставщика из числа других, определить и указать нужную номенклатуру из числа товаров или добавить новую запись. Пункт «Расход товара» подразумевает выписку товара со склада по документам, в виде расходной накладной, соответственно и ее создание. Также очередной функцией модуля выступает формирование прайс-листов, несущих в себе всю информацию об имеющемся товаре, включая его параметры (цена, состояние и т.д.).

Модуль «Формирование отчетов» выполняет функции по формированию печатных форм. Модуль «Отчеты» формируется, используя запросы, которые обрабатывают изначальные данные исходя из заданных пользователем параметров. В качестве информационной базы модуля «Формирование отчетов» выступают таблицы «Клиенты», «Товары», «Поставщики», «Приход товара», «Расход товара».

2.2. Обоснование выбора средств

Улучшение уровня управления системой и оперативности принятия решений стали основополагающими задачами в нашем вопросе повышения эффективности работы систем управления складским учетом. Результатами нашей работы должна была стать система, позволяющая своевременно решать появляющиеся задачи, и позволить предприятию, использующему складскую систему работать эффективно и стабильно, в том числе обеспечивающая работников предприятия удобной и понятной системой интерфейса и решений, что призваны облегчить работу персонала в данном вопросе.

А именно: разгрузить работников переложив работу с документацией и товарным движением на программу, тем самым произведя процесс автоматизации работы предприятия. В свою очередь программа должна быть информационно защищена от потери и кражи ценных данных предприятия, имея при себе необходимую структуру.

Рассмотрев перечисленные выше программные продукты конкурентов, представленные на рынке было принято решение о создании собственного продукта, учитывающего все проблемы и ошибки конкурентов и предыдущих систем складского учета. В качестве основного инструмента создания была выбрана система «1С: Предприятие».

Данный программный продукт является творением российской компании 1С, созданной в 1991 году, и занимающейся разработкой и поддержкой компьютерное ПО как для домашнего, так и для делового

сегмента пользователей [3]. Первоначально предназначенная для автоматизации бухгалтерского учета программа, на данный момент представляет из себя технологичную платформу, включающую в себя как режим работы в качестве разработчика, так и пользовательский режим, который по ошибке многими считается единственным в продуктах данной компании [4]. Платформа предоставляет алгоритмы управления данными и метаданными и позволяет создавать и изменять конфигурации работы с ними. Объекты платформы представлены в виду конфигураций, с которыми разработчик может взаимодействовать во время процесса автоматизации [6].

В процессе создания продукта для организации создается собственная конфигурация объектов, направленная на конкретное предприятие и его задачи. Данная конфигурация представляет собой законченное прикладное решение.



Рисунок 2.2.1 - Пирамида программных решений

Процесс создания проходит в специальной модели работы программного обеспечения, именуемого Конфигуратором. «Конфигуратор» - предметно-ориентированная среда разработки, входящая в состав платформы «1С:Предприятие 8» [7]. Среда предоставляет разработчику интегрированный инструментарий, требуемых для детальной разработки средств, распространения и поддержки прикладного решения для автоматизации работы предприятия. В дополнение к этому разработчик получает в распоряжение ряд инструментов, не имеющих непосредственного отношения к конфигуратору, но также призванных сделать труд разработчика более продуктивным [9].

Технологическая платформа в структуре «1С:Предприятие» (рисунок 1.6.1) - программная оболочка находящаяся над БД (базы на основе DBF-файлов в 7.7, эксклюзивный формат 1CD начиная от версии 8.0 или СУБД Microsoft SQL Server) [11]. Также, начиная с серии 8.1 хранение информации

доступно в СУБД PostgreSQL и IBM DB2, а с программной серии 8.2 добавилась и Oracle [12].

Среда разработки «1С: Предприятие» обладает собственным языком программирования, используемый для доступа и взаимодействия с данными, также комплекс действий с ПО инструментами, как OLE и DDE, в версиях 7.7, 8.0 и 8.1, посредством соединения формата COM [14]. На данный момент система функционирует лишь в операционной системе Windows. Однако начиная с программ номерного ряда 8.1, сервера в формате работы Клиент-Сервер «1С: Предприятия» могут работать на ОС LINUX [14].

Следующим программным средством, требующим выбора становится СУБД. К этому процессу необходимо подходить осмысленно и с должной ответственностью, не теряя из виду все важные факторы работы с ними и их особенности. Был проведен разговор с опытными специалистами в данной сфере, тем самым прибегнув к обширной базе опыта профессионалов данной отрасли, имевших практическую часть работы с многочисленными системами управления базами данных, как в бесплатных вариациях, так и в платных эксклюзивах. Важным факторов отбора будут персональные отличия вашей фирмы, а именно количество пользователей, что будут взаимодействовать с программой, требуемый функционал и так далее. Практика субъектов, взаимодействовавших с данными продуктами в процессе работы над 1С показывает, что продукты компаний Microsoft и Oracle намного производительнее их бесплатных аналогов. В качестве минусов бесплатных программ можно привести:

- 1) Сохранение памяти на будущее, тем самым увеличивая вес базы данных.
- 2) Слабые возможности механизмов взаимодействия.
- 3) Малое наполнение шаблонами, и в некоторых моментах полное их отсутствие и невозможность использования.

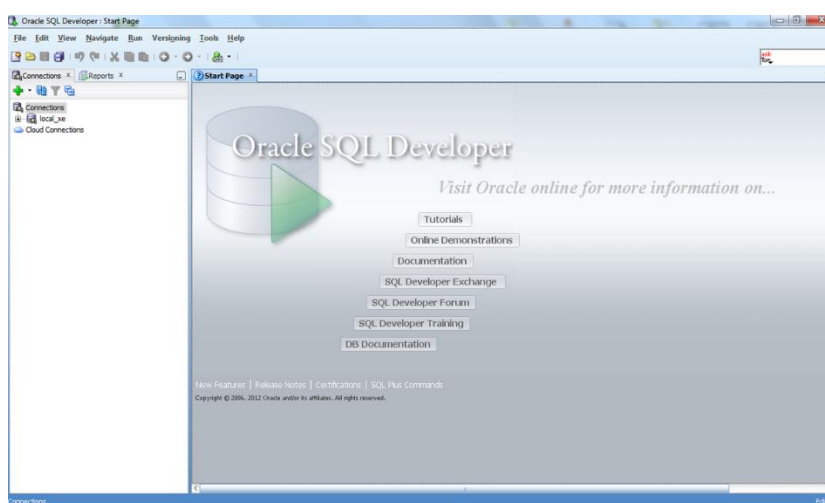


Рисунок 2.2.2 – Интерфейс Oracle SQL

В качестве первой рассматриваемой системы возьмем СУБД ORACLE (рисунок 2.2.2.) выпускаемой одноименной американской фирмой Oracle. В качестве плюсов приведем лояльную систему переноса приложения на другие ОС, высокую скорость работы даже в максимально затратных фрагментах работы с большим количеством разнообразной работы от разных пользователей и возможность локального вмешательства, без необходимости отключать всю программу, ради решения отдельно стоящей задачи. Но в то же время данная система довольно требовательна в вопросе квалификации администраторов, что забирает у данного решения пару баллов.

В итоге взгляд пал на самое популярное на рынке на данный момент решение от компании Microsoft, а именно MS SQL. Выгодными отличительными качествами данной системы стал алгоритм увеличения места по мере поступления данных и низкая требовательность к квалификации администраторского состава работников [15]. Но данная СУБД обладает высокими требованиями к ресурсам и всё-таки предлагает лишь платную лицензию.

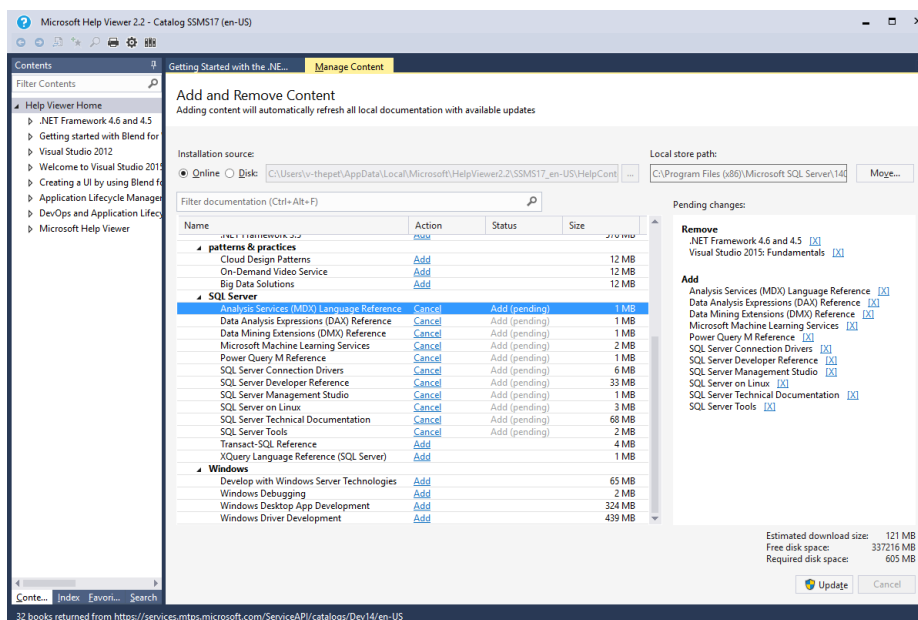


Рис 2.2.3 – Интерфейс MS SQL

Система управления реляционными базами данных (РСУБД), созданная компанией Microsoft. В качестве основного языка запросов представлен – Transact-SQL, произведенный в результате совместной работы «Microsoft» и «Sybase». Transact-SQL представляет собой реализацию такого стандарта как ANSI/ISO исходя из структурированного языка типа запросов (SQL) с дополнениями. Находит применение с базами данных различных размеров от учебных персональных до крупных производственных баз глобальных

корпораций; Удерживает лидирующие позиции в данном сегменте рынка, конкурируя с другими системами от сторонних производителей.

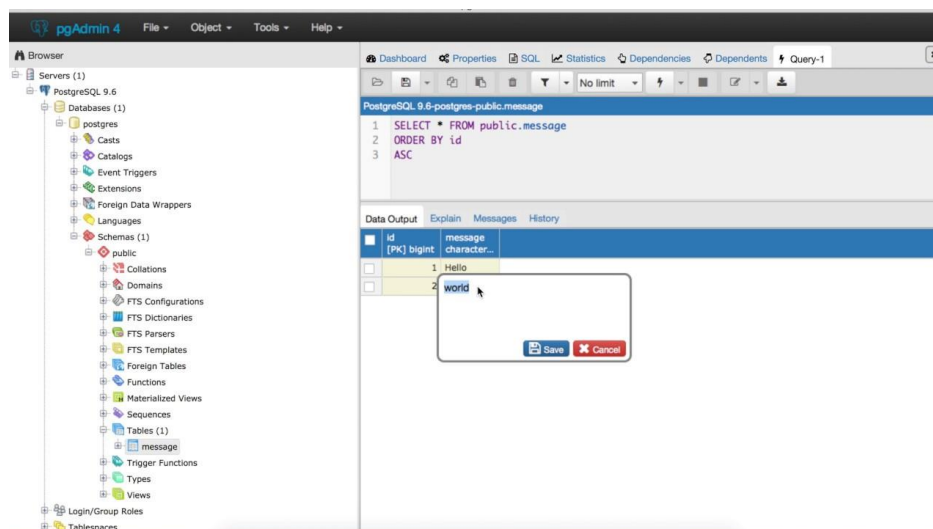


Рис 2.2.4 – Интерфейс PostgreSQL

В качестве инструмента проектирования баз данных в данном дипломном проекте была избрана СУБД PostgreSQL. PostgreSQL – является одним из популярных решений в вопросе проектировки SQL СУБД среди нынешнего поколения программистов. Местом разработки данной программы можно считать технический университет штата Калифорния, разработавший данное ПО в 1997 году. Подход к созданию и проектировке баз данных в данной программе считается новаторским в широких кругах разработчиков.

Решения, созданные для работы с информацией в данной СУБД являлись новшествами, и порадовали пользователей других СУБД наличием чего-то схожего лишь позднее. К слову данная СУБД является продуктом с открытой базой кода, что значительно ускоряет процесс ее развития, ввиду участия фан-базы в развитии и доработке.

Основополагающим фактором при выборе СУБД для нашего дипломного проекта была лицензия, а именно то, на каких условиях предоставлялась правообладателями. И именно наличие свободной лицензии в данном продукте повлияло на наш выбор, в том моменте, что открыта к корректировке функционала приложения под нужды конкретного пользователя. Несмотря на малую известность в широких кругах лиц, не занимающихся программированием, данная программа является поистине мощным и качественным инструментом для работы. Наличие бесплатной лицензии очень важным критерий отбора, и в данном случае один из камней преткновения в выборе, ибо большая часть платного ПО и решений на рынке стоят немалых денег, что создает немало проблем для начинающих предприятий ограниченных в вопросе финансов.

Часто бесплатные СУБД страдают от низкокачественной синхронизации во взаимодействии с форматами баз данных, но PostgreSQL является исключением.

Детали в работе данной СУБД несомненно важны, и наличие современных инструментов в данном случае не является слабым местом, программа обладает такими возможностями, как: external key, request-system, гибкие вариации показа, инструменты реагирования и т. п.

В списке эксклюзивных возможностей работы с PostgreSQL числится инструмент создания своих элементов взаимодействия с информацией. К примеру, создание собственной функции, свои операнды, функции агрегации, усложненная система индексов, и индивидуально созданные типы данных – все перечисленные выше действия возможны в данной СУБД.

Архитектура PostgreSQL функционирует в формате клиент-сервер. То есть, имеется главный сервер, так называемый postgres. В его задачи входит работа с запросами и файлами БД. Эта совокупность действий может функционировать как на компьютере клиента, так и непосредственно на серверной базе. Серверный процесс имеет мультизадачность и может работать с большим числом клиентов. На протяжении всего процесса серверное подключение доступно для клиентов.

Клиентское же приложение, что именуется в данной схеме, клиентом, имеет доступ к базе данных для взаимодействия в виде запросов. Приложения данного типа довольно разнообразны: утилиты текстового взаимодействия, графического взаимодействия, и другие направленные для работы с базой данных.

Наш выбор в вопросе СУБД будет перспективным решением для развития нашего комплекса складского учета. Объяснить это можно довольно просто, данное программное решение, имеющее перспективу и задел на будущее, ввиду полноценной поддержки, гибкости и открытости в вопросе улучшения качеств работы, благодаря открытому коду и большому числу лиц, вовлеченному в данный процесс, тем самым есть меньшая вероятность остановки действия данной СУБД и последующей нужды перевода всей базы на другую систему управления, что несомненно займет немало времени и ресурсов, также имея уязвимости в виде утечек информации предприятия.

3 Разработка программного комплекса

В данной главе будет рассмотрена предметную область складского предприятия, смоделировать бизнес-процессы используя средство для визуального моделирования бизнес-процессов BPWin, разработать базу данных предприятия, создать связи для таблиц, создать функциональную часть приложения и её интерфейс.

Предметная область – совокупность объектов, требуемая для пользователей, непосредственно взаимодействующих с конечным продуктом.

Предметной областью программного комплекса складского учета для ТОО “Restofood”, является организация функционального набора склада и упрощение работы складского персонала. В список персонала, использующему конечный продукт являются: заведующий складского помещения, отдел поставок, отдел сбыта, в обязанности которых входят работа с документацией и ведение складского товара. Тем самым, программный комплекс должен быть хранилищем для большого количества данных, добытых из документации и бухгалтерских номенклатур (производители, клиентура, товарная база), а также создавать и вести документационную часть (накладная, журнал складского учета, лист инвентаризации).

В цели автоматизации склада входят 3 категории: информационная складская часть, товарный учет и товарная аналитика.

В работу информационно-складской части входит контроль записей в справочной части, а именно: поставщики, работники, товары, клиенты, а также их редактирование.

Учет представляет собой ведение товарной логистика склада, приход товаров на склад и отпуск товара клиентуре. Начальным входным документом для склада является накладная товарного прихода, в которой указаны количество товара, его названия, типы и производители. Документом отпуска товара является накладная отпуска товара, с указанными в ней параметрами отпускаемого товара (наименования, вес, стоимость, а также данные покупателя, общая сумма отпускаемого товара).

Товарный анализ представляет собой систему запросов, требуемых данных. Может предполагать разную информацию о взаимодействиях склада, а именно о клиентуре, поставщиках, покупателей, работников.

По запросу программа может предоставить такую информацию, как:

- товарный список;
- Данные заказа клиента;
- сортированная информация о поставщиках;
- товарная история за отрезок времени;
- зарплатные ведомости;
- данные по работе сотрудников;

- продажи по сотрудникам;
- сортированные продажи (по клиентам, поставщикам);

Проведя анализ всей информации, требуется создание схемы информации нашей предметной области склада отдела Ногеса.

3.1 Создание кода

Итак, предприятие ожидает, что наш программный продукт сможет вобрать в себя лучшие стороны аналогов представленных на рынке, избавившись от недостатков. В данном моменте, стоит заметить, что для предприятия, работающего в сегменте Ногеса большая часть функций аналогов будет излишней и лишь послужит причиной перегрузки технического оборудования предприятия.

Начнем с модуля «Продажи». Данный модуль в типовых конфигурациях аналогов представляет собой огромный комплекс опций, в т.ч. подсистемы казначейства, кассовые книги, приходной кассового дня, оплата в кредит и т.д. Для работы разрабатываемого продукта вышеописанные опции будут излишни, создавая ненужную предприятию информацию.

Данный модуль состоит из следующих разделов:

Выписка банка – Раздел, созданный для ведения, редактирования и полного контроля над расходными взаимодействиями с банком:

Предприятие предоставляет информацию о денежных движениях банку, данная информация должна находиться в выписках, содержащих данные о приходе и расходе денежных средств с контрагентами.

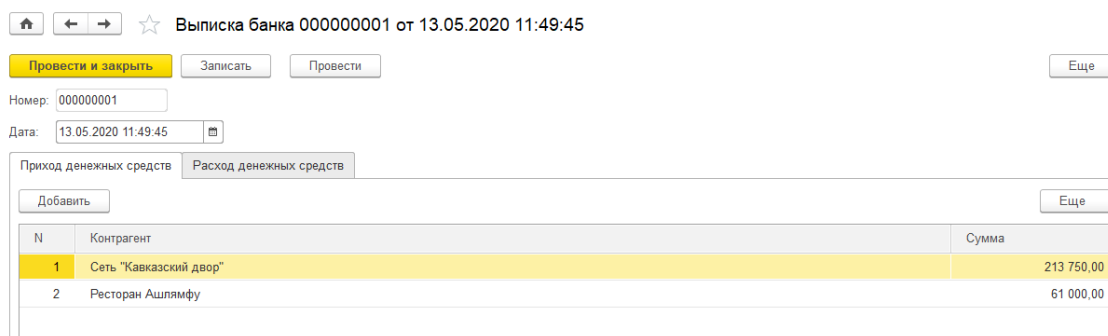
```

Процедура ОбработкаПроведения (Отказ, Режим)
    Движения.Взаиморасчеты.Записывать = Истина;
    Движения.РегистрБухУчет.Записывать = Истина;
    Для Каждого ТекСтрокаПриходДенежныхСредств Из ПриходДенежныхСредств Цикл
        // регистр Взаиморасчеты Расход
        Движение = Движения.Взаиморасчеты.Добавить ();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Контрагент = ТекСтрокаПриходДенежныхСредств.Контрагент;
        Движение.Сумма = ТекСтрокаПриходДенежныхСредств.Сумма;
        // регистр РегистрБухУчет
        Движение = Движения.РегистрБухУчет.Добавить ();
        Движение.СчетДт = ПланыСчетов.БухгалтерскийУчет.РасчетныйСчет;
        Движение.СчетКт = ПланыСчетов.БухгалтерскийУчет.Покупатели;
        Движение.Период = Дата;
        Движение.Сумма = ТекСтрокаПриходДенежныхСредств.Сумма;
    КонечЦикла;
    Для Каждого ТекСтрокаРасходДенежныхСредств Из РасходДенежныхСредств Цикл
        // регистр Взаиморасчеты Приход
        Движение = Движения.Взаиморасчеты.Добавить ();
        Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
        Движение.Период = Дата;
        Движение.Контрагент = ТекСтрокаРасходДенежныхСредств.Контрагент;
        Движение.Сумма = ТекСтрокаРасходДенежныхСредств.Сумма; // регистр РегистрБухУчет
        Движение = Движения.РегистрБухУчет.Добавить ();
        Движение.СчетДт = ПланыСчетов.БухгалтерскийУчет.Поставщики;
        Движение.СчетКт = ПланыСчетов.БухгалтерскийУчет.РасчетныйСчет;
        Движение.Период = Дата;
        Движение.Сумма = ТекСтрокаРасходДенежныхСредств.Сумма;
    КонечЦикла;
КонечПроцедуры

```

Рисунок 3.1.1 – Код «Выписка банка»

Участок кода проводит запись необходимых данных в файлы, требуемые для последующей аналитики и действий. Также записывает необходимую информацию в поля бухгалтерии.



Выписка банка 000000001 от 13.05.2020 11:49:45

Провести и закрыть Записать Провести Еще

Номер: 000000001

Дата: 13.05.2020 11:49:45

Приход денежных средств Расход денежных средств

Добавить Еще

N	Контрагент	Сумма
1	Сеть "Кавказский двор"	213 750,00
2	Ресторан Ашлямфу	61 000,00

Рисунок 3.1.2 – Выписка банка

В итоговой выписке должна быть представлена требуемая информация о контрагенте (денежная сумма прихода или расхода, и наименование контрагента).

Заказ клиента – Раздел для работы с накладными, заказами клиента, с указанием всех требуемых полей (данные о клиенте, сотруднике проводящем операцию, складе, статусе сделки, и также имеет идентификатор заказа):

В работе предприятия отсутствует какая-либо система скидок, исходя из этого возможность добавления скидки в программе отсутствует, следовательно, расчет итоговой суммы проводится по подсчету цены и количества товара.

```

□ Процедура ОбработкаПроведения(Отказ, Режим)
  // регистр ЭффективностьПродаж
  Движения.ЭффективностьПродаж.Записывать = Истина;
  Движение = Движения.ЭффективностьПродаж.Добавить();
  Движение.Период = Дата;
  Движение.Сотрудник = Сотрудник;
  Движение.КоличествоЗаказов = 1;
  Запрос = Новый Запрос;
  Запрос.Текст =
    "ВЫБРАТЬ
     | СтатусыСделок.Ссылка
     |ИЗ
     | Справочник.СтатусыСделок КАК СтатусыСделок
     |ГДЕ
     | СтатусыСделок.НомерПоПорядку <= &НомерПоПорядку";
  Запрос.УстановитьПараметр("НомерПоПорядку", СтатусСделки.НомерПоПорядку);
  РезультатЗапроса = Запрос.Выполнить();
  ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
  Движения.ВоронкаПродаж.Записывать = Истина;
  Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Движение = Движения.ВоронкаПродаж.Добавить();
    Движение.Период = Дата;
    Движение.СтатусСделки = ВыборкаДетальныеЗаписи.Ссылка;
    Движение.КоличествоСделок = 1;
    Движение.Сумма = СуммаДокумента;
  КонецЦикла;
КонецПроцедуры
□ Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)
  СуммаДокумента = Товары.Итог("Сумма");
КонецПроцедуры

```

Рисунок 3.1.3 – Код «Заказ Клиента»

Код проводит заказ клиента, учитывая все поля, сопутствующие заказу, которые впоследствии будут использованы для описания заказа, и дальнейшего облегчения идентификации.

В вышеописанном фрагменте кода проводятся действия связанные с клиентскими заказами, не только в моменте создания заявки, но также внесения данных для статистики (воронка продаж), и создание счета оплаты для клиента без лишней информации.

Счет на оплату

Номер 000000002
 Дата 13.05.2020 11:49:38
 Клиент Ресторан Ашлямфу

№	Товар	Количество	Цена	Сумма
1	Мандарины Эквадор	30,000	800,00	24 000,00
2	Малина	10,000	1 100,00	11 000,00
3	Папайя	10,000	2 600,00	26 000,00

Бубликов В.В. _____

Склад Склад Зенкова 25

Рисунок 3.1.4 – Счет на оплату модуля «Заказ клиента»

На склад предприятия и в базу необходимо поступление товаров, следующую информацию необходимо вносить в приложение для последующих операций с её использованием. Товар должен обладать своими полями-идентификаторами, срок годности товара не добавлен, ввиду желания предприятия вести контроль вживую. Также в данном фрагменте будет внесена информация о себестоимости и поставщиках, требуемая для последующего подсчета прибыли и прочих документаций.

```

Процедура ОбработкаПроведения(Отказ, Режим)
  {{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
  // регистр Взаиморасчеты Расход
  Движения.Взаиморасчеты.Записывать = Истина;
  Движение = Движения.Взаиморасчеты.Добавить ();
  Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
  Движение.Период = Дата;
  Движение.Контрагент = Поставщик;
  Движение.Сумма = СуммаДокумента;
  Движения.ТоварыНаСкладах.Записывать = Истина;
  Движения.СебестоимостьТоваров.Записывать = Истина;
  Для Каждого ТекСтрокаТовары Из Товары Цикл
    // регистр ТоварыНаСкладах Приход. Приход товара на склад и внесение базовых полей.
    Движение = Движения.ТоварыНаСкладах.Добавить ();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Номенклатура = ТекСтрокаТовары.Товар;
    Движение.Склад = Склад;
    Движение.Количество = ТекСтрокаТовары.Количество;
    // регистр СебестоимостьТоваров Приход
    Движение = Движения.СебестоимостьТоваров.Добавить ();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Номенклатура = ТекСтрокаТовары.Товар;
    Движение.Сумма = ТекСтрокаТовары.Сумма;
    Движение.Количество = ТекСтрокаТовары.Количество;
  КонецЦикла;
  // регистр РегистрБухУчет. Внесение данной информации в использование модулю бухгалтерия.
  
```

Рисунок 3.1.5 – Код «Поступление товаров и услуг»

```

// регистр РегистрБухУчет. Внесение данной информации в использование модулю бухгалтерия.
Движения.РегистрБухУчет.Записывать = Истина;
Движение = Движения.РегистрБухУчет.Добавить ();
Движение.СчетДт = ПланыСчетов.БухгалтерскийУчет.Товары;
Движение.СчетКт = ПланыСчетов.БухгалтерскийУчет.Поставщики;
Движение.Период = Дата;
Движение.Сумма = СуммаДокумента;
  }}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)
  СуммаДокумента = Товары.Итог("Сумма");
КонецПроцедуры
  
```

Рисунок 3.1.6 – Код «Поступление товаров и услуг»

Код проводит запись информации и вносит ее в документы, в последствии использующие данные о товарном поступлении в складском и бухгалтерском учете.

Теперь программа обладает информацией о имеющемся товаре, который предприятие может реализовать в следующем модуле.

Следующим является «Реализация товаров и услуг». В данном разделе отражается факт продажи товаров или услуг.

Начнем создание процедуры внесения данных. В финальном интерфейсе информация вводится построчно.

```
□ Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)
  //{{__КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
  Если ТипЗнч(ДанныеЗаполнения) = Тип("ДокументСсылка.ЗаказКлиента") Тогда
    // Заполнение шапки
    Клиент = ДанныеЗаполнения.Клиент;
    Склад = ДанныеЗаполнения.Склад;
    Для Каждого ТекСтрокаТовары Из ДанныеЗаполнения.Товары Цикл
      НоваяСтрока = Товары.Добавить();
      НоваяСтрока.Количество = ТекСтрокаТовары.Количество;
      НоваяСтрока.Сумма = ТекСтрокаТовары.Сумма;
      НоваяСтрока.Товар = ТекСтрокаТовары.Товар;
      НоваяСтрока.Цена = ТекСтрокаТовары.Цена;
    КонечЦикла;
  КонечЕсли;
  //}}__КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
КонецПроцедуры
```

Рисунок 3.1.7 – Обработка заполнения «Реализация Товаров и Услуг»

В данном фрагменте кода проводится внесение основных данных по реализованным товарам и услугам.

N	Товар	Количество	Цена	Сумма
1	Яблоко Салтанат	30,000	500,00	15 000,00
2	Картофель	250,000	200,00	50 000,00
3	Укроп	50,000	120,00	6 000,00
4	Морковь	100,000	300,00	30 000,00
5	Малина	25,000	800,00	20 000,00

Рисунок 3.1.8 – Пример внесенных данных о реализуемом товаре.

Следующий фрагмент кода будет участком обработки вносимой информации и последующего её внесения в документацию.


```

□ Процедура ОбработкаПроведения(Отказ, Режим)
  // регистр Взаиморасчеты Приход, запись общей информации о
  // приходе в определенный момент времени, от конкретного контрагента.
  Движения.Взаиморасчеты.Записывать = Истина;
  Движение = Движения.Взаиморасчеты.Добавить ();
  Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
  Движение.Период = Дата;
  Движение.Контрагент = Клиент;
  Движение.Сумма = СуммаДокумента;
  // регистр ТоварыНаСкладах Расход, в данном фрагменте вносятся
  // изменения в товарные составляющие, с учетом реализованного товара.
  Движения.ТоварыНаСкладах.Записывать = Истина;
  Для Каждого ТекСтрокаТовары Из Товары Цикл
    Движение = Движения.ТоварыНаСкладах.Добавить ();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Номенклатура = ТекСтрокаТовары.Товар;
    Движение.Склад = Склад;
    Движение.Количество = ТекСтрокаТовары.Количество;
  КонецЦикла;
  Движения.Записать ();
  Если Режим = РежимПроведенияДокумента.Оперативный Тогда
    Запрос = Новый Запрос;
    Запрос.Текст =
      "ВЫБРАТЬ
      |   ТоварыНаСкладахОстатки.Номенклатура,
      |   ТоварыНаСкладахОстатки.Склад,
      |   -ТоварыНаСкладахОстатки.КоличествоОстаток КАК Количество
      |ИЗ
  
```

Рисунок 3.1.9 – Обработка проведения «Реализация Товаров и Услуг»

```

|ИЗ
|   РегистрНакопления.ТоварыНаСкладах.Остатки(
|       ,
|       Склад = &Склад
|       И Номенклатура В
|       (ВЫБРАТЬ
|           РеализацияТоваровИУслугТовары.Товар
|           ИЗ
|           Документ.РеализацияТоваровИУслуг.Товары КАК РеализацияТоваровИУслугТовары
|           ГДЕ
|           РеализацияТоваровИУслугТовары.Ссылка = &Ссылка)) КАК ТоварыНаСкладахОстатки
|ГДЕ
|   ТоварыНаСкладахОстатки.КоличествоОстаток < 0";
Запрос.УстановитьПараметр("Склад", Склад);
Запрос.УстановитьПараметр("Ссылка", Ссылка);
РезультатЗапроса = Запрос.Выполнить ();
Если НЕ РезультатЗапроса.Пустой() Тогда
  Отказ = Истина;
  ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать ();
  Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Сообщить("Недостаточно товара "+ВыборкаДетальныеЗаписи.Номенклатура+" в количестве "
    +ВыборкаДетальныеЗаписи.Количество);
  КонецЦикла;
КонецЕсли;
Если Отказ Тогда
  Возврат;
КонецЕсли;
Движения.СебестоимостьТоваров.Записывать = Истина;
  
```

Рисунок 3.1.10 – Обработка проведения «Реализация Товаров и Услуг»

```

Движения.СебестоимостьТоваров.Записывать = Истина;
// регистр Продажи, в данном фрагменте рассчитывается прибыль
// полученная предприятием, исходя из подсчета разницы
// себестоимости и итоговой цены реализации.
// А также вносится информация, после используемая в графиках статистики.
Движения.Продажи.Записывать = Истина;
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
| СебестоимостьТоваровОстатки.Номенклатура,
| СебестоимостьТоваровОстатки.СуммаОстаток КАК Сумма,
| СебестоимостьТоваровОстатки.КоличествоОстаток КАК Количество
|ИЗ
| РегистрНакопления.СебестоимостьТоваров.Остатки (
|     &МоментВремени,
|     Номенклатура В
|     (ВЫБРАТЬ
|         РеализацияТоваровИУслугТовары.Товар
|     ИЗ
|         Документ.РеализацияТоваровИУслуг.Товары КАК РеализацияТоваровИУслугТовары
|     ГДЕ
|         РеализацияТоваровИУслугТовары.Ссылка = &Ссылка) КАК СебестоимостьТоваровОстатки";
Запрос.УстановитьПараметр("МоментВремени", МоментВремени());
Запрос.УстановитьПараметр("Ссылка", Ссылка);
РезультатЗапроса = Запрос.Выполнить();
ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
СуммаСебестоимости = 0;
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Если ВыборкаДетальныеЗаписи.Количество <> 0 Тогда
        СебестоимостьЕдиницы = ВыборкаДетальныеЗаписи.Сумма/ВыборкаДетальныеЗаписи.Количество;

```

Рисунок 3.1.11 – Обработка проведения «Реализация Товаров и Услуг»

```

        СебестоимостьЕдиницы = ВыборкаДетальныеЗаписи.Сумма/ВыборкаДетальныеЗаписи.Количество;
Иначе
    СебестоимостьЕдиницы = 0;
КонецЕсли;
Движение = Движения.СебестоимостьТоваров.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
СтрокаТЧ = Товары.Найти(ВыборкаДетальныеЗаписи.Номенклатура, "Товар");
Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Количество = СтрокаТЧ.Количество;
СебестоимостьСписания = СебестоимостьЕдиницы * СтрокаТЧ.Количество;
Движение.Сумма = СебестоимостьСписания;
// Движения по регистру Продажи.
// фрагмент вносит записи, в последующем используемые модулем бухгалтерии.
Движение = Движения.Продажи.Добавить();
Движение.Период = Дата;
Движение.Клиент = Клиент;
Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Количество = СтрокаТЧ.Количество;
Движение.Сумма = СтрокаТЧ.Сумма;
Движение.Себестоимость = СебестоимостьСписания;
СуммаСебестоимости = СуммаСебестоимости + СебестоимостьСписания;
КонецЦикла;
Движения.РегистрБухУчет.Записывать = Истина;
Проводка = Движения.РегистрБухУчет.Добавить();
Проводка.Период = Дата;
Проводка.СчетДт = ПланыСчетов.БухгалтерскийУчет.Покупатели;
Проводка.СчетКт = ПланыСчетов.БухгалтерскийУчет.Выручка;
Проводка.Сумма = СуммаДокумента;
Проводка = Движения.РегистрБухУчет.Добавить();

```

Рисунок 3.1.12 – Обработка проведения «Реализация Товаров и Услуг»

```

Проводка = Движения.РегистрБухУчет.Добавить ();
Проводка.Период = Дата;
Проводка.СчетДт = ПланыСчетов.БухгалтерскийУчет.Себестоимость;
Проводка.СчетКт = ПланыСчетов.БухгалтерскийУчет.Товары;
Проводка.Сумма = СуммаСебестоимости;
Движения.ЭффективностьПродаж.Записывать = Истина;
Движение = Движения.ЭффективностьПродаж.Добавить ();
Движение.Период = Дата;
Движение.Сотрудник = Сотрудник;
Движение.КоличествоОтгрузок = 1;
Движение.Выручка = СуммаДокумента;
Движение.Прибыль = СуммаДокумента - СуммаСебестоимости;
КонецПроцедуры
Процедура ОбработкаПроверкиЗаполнения(Отказ, ПроверяемыеРеквизиты)
    // Вставить содержимое обработчика.
КонецПроцедуры
Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)
    СуммаДокумента = 0;
    Для Каждого СтрокаТЧ ИЗ Товары Цикл
        СуммаДокумента = СуммаДокумента + СтрокаТЧ.Сумма;
        //Сообщить (СтрокаТЧ.Товар.ОсновнойПоставщик);
    КонецЦикла;
    //Сообщить (МоментВремени ());
КонецПроцедуры

```

Рисунок 3.1.13 – Обработка проведения «Реализация Товаров и Услуг»

Реализация товаров и услуг 000000002 от 12.05.2020 0:00:00 *

Основное | Взаиморасчеты | Продажи | Себестоимость товаров | Товары на складах

Провести и закрыть | Записать | Провести | Еще

Дата: 12.05.2020 0:00:00

Клиент: Ресторан Ашлямфу

Склад: Склад Зенкова 25

Сотрудник: Зареченская В.Н.

Добавить | Еще

N	Товар	Количество	Цена	Сумма
1	Яблоко Салтанат	30,000	500,00	15 000,00
2	Картофель	250,000	200,00	50 000,00
3	Укроп	50,000	120,00	6 000,00
4	Морковь	100,000	300,00	30 000,00
5	Малина	25,000	800,00	20 000,00
6				

Рисунок 3.1.14 – Заполнение в модуле «Реализация товаров и услуг»

После полного заполнения документ можно провести. При проведении сформируются проводки, которые отразят факт реализации товара в бухгалтерском учете.

И один из важнейших разделов для работы предприятия. Раздел «Товары»:

Товары должны обладать полями для их идентификации, а также находиться в доступе для модулей реализации и поступления.

После внесения товара на склад, товар должен получить цены на номенклатуры. То есть быть внесенным в базу с данными под реализацию.

```

□ Процедура ОбработкаПроведения(Отказ, Режим)
  //{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!

  // регистр ЦеныНоменклатуры
  Движения.ЦеныНоменклатуры.Записывать = Истина;
  Для Каждого ТекСтрокаЦены Из Цены Цикл
    Движение = Движения.ЦеныНоменклатуры.Добавить ();
    Движение.Период = Дата;
    Движение.Номенклатура = ТекСтрокаЦены.Номенклатура;
    Движение.Цена = ТекСтрокаЦены.Цена;
  КонечЦикла;

  //}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  КонечПроцедуры

```

Рисунок 3.1.15 – Код «Установка цен на номенклатуры»

С учетом всех полученных данных ведется запись для бухгалтерской документации. Общий алгоритм записи или удаления данных для бухгалтерских операций следующий:

```

□ Процедура ОбработкаПроведения(Отказ, Режим)
  //{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!

  // регистр ЦеныНоменклатуры
  Движения.ЦеныНоменклатуры.Записывать = Истина;
  Для Каждого ТекСтрокаЦены Из Цены Цикл
    Движение = Движения.ЦеныНоменклатуры.Добавить ();
    Движение.Период = Дата;
    Движение.Номенклатура = ТекСтрокаЦены.Номенклатура;
    Движение.Цена = ТекСтрокаЦены.Цена;
  КонечЦикла;

  //}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  КонечПроцедуры

```

Рисунок 3.1.16 – Код «Бухгалтерская операция»

Бухгалтерская операция «Закрытие месяца» несомненно важна для предприятий торговых сфер деятельности.

```

□ функция ОстатокПоСчету(Счет, Период)
  Запрос = Новый Запрос;
  Запрос.Текст =
    "ВЫБРАТЬ
    | РегистрБухУчетОстатки.СуммаОстатокДт,
    | РегистрБухУчетОстатки.СуммаОстатокКт
    |ИЗ
    | РегистрБухгалтерии.РегистрБухУчет.Остатки(&Период, Счет = &Счет, ) КАК РегистрБухУчетОстатки";
  Запрос.УстановитьПараметр("Период", Период);
  Запрос.УстановитьПараметр("Счет", Счет);
  РезультатЗапроса = Запрос.Выполнить();
  Если РезультатЗапроса.Пустой() Тогда
    Возврат 0;
  КонечЕсли;
  ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
  ВыборкаДетальныеЗаписи.Следующий();
  Если Счет.Вид = ВидСчета.Активный Тогда
    Возврат ВыборкаДетальныеЗаписи.СуммаОстатокДт;
  Иначе
    Возврат ВыборкаДетальныеЗаписи.СуммаОстатокКт;
  КонечЕсли;
  Конечфункции

```

Рисунок 3.1.17 – Код «Закрытие месяца»

В данном участке кода устанавливаются данные по запросу пользователя. А также ведет расчет по бухгалтерским показателям.

```
□ Процедура ОбработкаПроведения(Отказ, РежимПроведения)
  КонецПериода = КонецМесяца(Дата)+1;
  ОстатокВыручка = ОстатокПоСчету(ПланыСчетов.БухгалтерскийУчет.Выручка, КонецПериода);
  ОстатокСебестоимость = ОстатокПоСчету(ПланыСчетов.БухгалтерскийУчет.Себестоимость, КонецПериода);
  Движения.РегистрБухУчет.Записывать = Истина;
  Проводка = Движения.РегистрБухУчет.Добавить();
  Проводка.Период = Дата;
  Проводка.СчетДт = ПланыСчетов.БухгалтерскийУчет.Выручка;
  Проводка.СчетКт = ПланыСчетов.БухгалтерскийУчет.Прибыль;
  Проводка.Сумма = ОстатокВыручка;
  Проводка = Движения.РегистрБухУчет.Добавить();
  Проводка.Период = Дата;
  Проводка.СчетДт = ПланыСчетов.БухгалтерскийУчет.Прибыль;
  Проводка.СчетКт = ПланыСчетов.БухгалтерскийУчет.Себестоимость;
  Проводка.Сумма = ОстатокСебестоимость;
  КонецПроцедуры
```

Рисунок 3.1.18 – Код «Закрытие месяца»

Предприятию было необходимо получить программный продукт, объединяющий все нужные функции из разных программ. В складские комплексы обычно действия с работниками не входят, такие модули как правило выносятся в отдельные программы. Исходя из желаний заказчика, был добавлен полноценный модуль взаимодействия с работниками предприятия. Модуль позволяет вести работу с зарплатами, графиками, премиями, и другими важными составляющими взаимодействия с сотрудниками. Одним из первых рассмотрим фрагмент, отвечающий за графики работы сотрудника:

```
□ Процедура ОбработкаПроведения(Отказ, РежимПроведения)
  КонецПериода = КонецМесяца(Дата)+1;
  ОстатокВыручка = ОстатокПоСчету(ПланыСчетов.БухгалтерскийУчет.Выручка, КонецПериода);
  ОстатокСебестоимость = ОстатокПоСчету(ПланыСчетов.БухгалтерскийУчет.Себестоимость, КонецПериода);
  Движения.РегистрБухУчет.Записывать = Истина;
  Проводка = Движения.РегистрБухУчет.Добавить();
  Проводка.Период = Дата;
  Проводка.СчетДт = ПланыСчетов.БухгалтерскийУчет.Выручка;
  Проводка.СчетКт = ПланыСчетов.БухгалтерскийУчет.Прибыль;
  Проводка.Сумма = ОстатокВыручка;
  Проводка = Движения.РегистрБухУчет.Добавить();
  Проводка.Период = Дата;
  Проводка.СчетДт = ПланыСчетов.БухгалтерскийУчет.Прибыль;
  Проводка.СчетКт = ПланыСчетов.БухгалтерскийУчет.Себестоимость;
  Проводка.Сумма = ОстатокСебестоимость;
  КонецПроцедуры
```

Рисунок 3.1.19 – Код «Утверждение графика работ»

Теперь фрагмент, отвечающий за установку и начисление заработной платы сотрудника в необходимый период времени:

```

Процедура ОбработкаПроведения(Отказ, Режим)
// регистр ЗарботнаяПлата
Движения.ЗарботнаяПлата.Записывать = Истина;
Движение = Движения.ЗарботнаяПлата.Добавить();
Движение.Сторно = Ложь;
Движение.ВидРасчета = ПланыВидовРасчета.Начисления.Оклад;
Движение.ПериодДействияНачало = НачалоМесяца(Дата);
Движение.ПериодДействияКонец = КонецМесяца(Дата);
Движение.ПериодРегистрации = Дата;
Движение.Сотрудник = Сотрудник;
Движение.РасчетныеДанные = Сумма;
Движения.Записать();
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
| ЗарботнаяПлатаДанныеГрафика.РабочийДеньПериодДействия КАК Норма,
| ЗарботнаяПлатаДанныеГрафика.РабочийДеньФактическийПериодДействия КАК факт,
| ЗарботнаяПлатаДанныеГрафика.РасчетныеДанные
| ИЗ
| РегистрРасчета.ЗарботнаяПлата.ДанныеГрафика(Регистратор = &Регистратор) КАК ЗарботнаяПлатаДанныеГрафика";
Запрос.УстановитьПараметр("Регистратор", Ссылка);
РезультатЗапроса = Запрос.Выполнить();
Если НЕ РезультатЗапроса.Пустой() Тогда
ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
ВыборкаДетальныеЗаписи.Следующий();
Если ВыборкаДетальныеЗаписи.Норма <> 0 Тогда
Оклад = ВыборкаДетальныеЗаписи.РасчетныеДанные * ВыборкаДетальныеЗаписи.факт / ВыборкаДетальныеЗаписи.Норма;

```

Рисунок 3.1.20 – Код «Начисление оклада»

```

Оклад = ВыборкаДетальныеЗаписи.РасчетныеДанные * ВыборкаДетальныеЗаписи.факт / ВыборкаДетальныеЗаписи.Норма;
Движение.Сумма = Оклад;
Движения.ЗарботнаяПлата.Записать();
Движения.ДолгиПоЗарботнойПлате.Записывать = Истина;
Движение = Движения.ДолгиПоЗарботнойПлате.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Сотрудник = Сотрудник;
Движение.Сумма = Оклад;
Иначе
Сообщить("Неверно заполнен график!");
КонецЕсли;
КонецЕсли;
- КонецПроцедуры

```

Рисунок 3.1.21 – Код «Начисление оклада»

Рассмотрим фрагмент выплаты заработной платы сотруднику:

```

Процедура ОбработкаПроведения(Отказ, Режим)
//{{ _КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

// регистр ДолгиПоЗарботнойПлате Расход
Движения.ДолгиПоЗарботнойПлате.Записывать = Истина;
Для Каждого ТекСтрокаСотрудники Из Сотрудники Цикл
Движение = Движения.ДолгиПоЗарботнойПлате.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Сотрудник = ТекСтрокаСотрудники.Сотрудник;
Движение.Сумма = ТекСтрокаСотрудники.Сумма;
КонецЦикла;

//}} _КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры

```

Рисунок 3.1.22 – Код «Выплата заработной платы»

Также неотъемлемой частью поощрения работы сотрудника является премия в дополнение к основной заработной плате, система премий в предприятии процентная, выставляется в виде процента от заработной платы:

```

□ Процедура ОбработкаПроведения(Отказ, Режим)
  // регистр ЗаработнаяПлата
  Движения.ЗаработнаяПлата.Записывать = Истина;
  Движение = Движения.ЗаработнаяПлата.Добавить ();
  Движение.Сторно = Ложь;
  Движение.ВидРасчета = ВидРасчета;
  Движение.ПериодДействияНачало = НачалоМесяца(Дата);
  Движение.ПериодДействияКонец = КонецМесяца(Дата);
  Движение.ПериодРегистрации = Дата;
  Движение.БазовыйПериодНачало = БазовыйПериодНачало;
  Движение.БазовыйПериодКонец = КонецДня(БазовыйПериодОкончание);
  Движение.Сотрудник = Сотрудник;
  Движение.РасчетныеДанные = Процент;
  Движения.Записать ();
  Запрос = Новый Запрос;
  Запрос.Текст =
    "ВЫБРАТЬ
     | ЗаработнаяПлатаБазаЗаработнаяПлата.СуммаБаза,
     | ЗаработнаяПлатаБазаЗаработнаяПлата.РасчетныеДанные
    |ИЗ
     | РегистрРасчета.ЗаработнаяПлата.БазаЗаработнаяПлата(«Измерения, «Измерения, , Регистратор = «Ссылка)
     | КАК ЗаработнаяПлатаБазаЗаработнаяПлата";
  Измерения = Новый Массив;
  Измерения.Добавить ("Сотрудник");
  Запрос.УстановитьПараметр ("Измерения", Измерения);
  Запрос.УстановитьПараметр ("Ссылка", Ссылка);
  РезультатЗапроса = Запрос.Выполнить ();
  Если НЕ РезультатЗапроса.Пустой () Тогда

```

Рисунок 3.1.23 – Код «Расчет премии»

```

  Если НЕ РезультатЗапроса.Пустой () Тогда
  ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать ();
  ВыборкаДетальныеЗаписи.Следующий ();
  Премия = ВыборкаДетальныеЗаписи.СуммаБаза * ВыборкаДетальныеЗаписи.РасчетныеДанные / 100;
  Движение.Сумма = Премия;
  Движения.ЗаработнаяПлата.Записать ();
  Движения.ДолгиПоЗаработнойПлате.Записывать = Истина;
  Движение = Движения.ДолгиПоЗаработнойПлате.Добавить ();
  Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
  Движение.Период = Дата;
  Движение.Сотрудник = Сотрудник;
  Движение.Сумма = Премия;
  КонецЕсли;
КонецПроцедуры

```

Рисунок 3.1.24 – Код «Расчет премии»

Премии также вносятся в графики по работникам. Одним из основных графиков, которые предприятие несомненно хотело иметь, были диаграммы ERM, также именуемые диаграммами Ганта. В данном предприятии выносятся расчет по отработанному времени, и о факте оплаты данного времени труда.

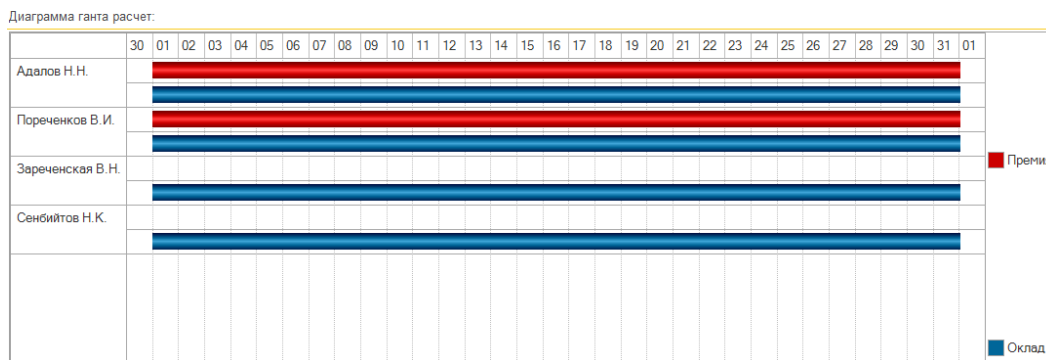


Рисунок 3.1.25 – Диаграмма Ганта по отработанному времени

В случае невыхода сотрудника на работу данные также должны быть внесены, для полноценной оценки рабочей службы, в случае прогула сумма зарплаты должна быть изменена, исходя из пропущенных без уважительной причины рабочих дней:

```

Процедура ОбработкаПроведения(Отказ, Режим)
//{{ __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

// регистр ЗарботнаяПлата
Движения.ЗарботнаяПлата.Записывать = Истина;
Движение = Движения.ЗарботнаяПлата.Добавить ();
Движение.Сторно = Ложь;
Движение.ВидРасчета = ПланыВидовРасчета.Начисления.Прогул;
Движение.ПериодДействияНачало = ДатаНачала;
Движение.ПериодДействияКонец = КонецДня(ДатаОкончания);
Движение.ПериодРегистрации = Дата;
Движение.Сотрудник = Сотрудник;

//}} __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры

```

Рисунок 3.1.26 – Код «Невыход сотрудника»

В итоге имеем программный продукт, лишенный всех излишних для предприятия функций, увеличивавших бы объемы заполнения информации на носителях. По сравнению с аналогами, представленными на рынке, в данном комплексе не были реализованы следующие опции:

- Подсистемы и отчеты казначейства;
- Заявки соглашения;
- Платежный календарь;
- Оплата в кредит;
- Приходные кассового дня;
- Подсистемы финансового результата;
- Подсистемы логистики Relog;
- Обмен данным Битрикс.

Следовательно, программный комплекс обеспечивая предприятие всеми необходимыми возможностями, не будет производить запись излишней информации, в последующем увеличивающей кэш программы до десятков гигабайт данных.

3.2 Системные требования

Для успешной работы программного комплекса предприятие должно быть оснащено оборудованием, а именно компьютером со следующими минимальными требованиями:

1) Минимум — двухъядерный процессор от 3 000 МГц и кэшем от 4 Мб. Оптимально: четырехъядерный процессор от 3 500 МГц и кэшем от 6 МБ.

2) Оперативная память для 32-битных систем одинакова и в минимальном, и оптимальном варианте, составляет 4 Гб. А для 64-битной требуется, как минимум 8 Гб, оптимально — 16 Гб.

3) Что касается жесткого диска, то минимальные требования: скорость 7200 при кэше от 64 Гб, SSD. Оптимально использовать только SSD жесткий диск.

При подборе технического оборудования советуем присмотреться к нашим приведенным ниже советам.

При выборе устройства памяти, важнейшей для нас характеристикой будет - скорость памяти, определяемая частотой процессора, поколением, количеством каналов, задержками и размером кэша.

Наименьшее внимание можете уделить такому параметру, как количество ядер. Теоретически для решения всех задач хватит и двухъядерного процессора, другое дело, что на практике процессор с двумя ядрами и режимом кратковременного ускорения не найти.

Еще один момент: SSD обеспечивает не только высокую скорость, но и очень короткое время доступа к данным. HDD могут компенсировать это системным или внутренним кэшем диска, но если речь идет о нагруженных системах, кэш будет просто не способен покрыть большие объемы, что в результате приведет к снижению скорости записи или считывания информации. Особенно актуальна эта проблема, если работать в нескольких базах: так, если одна обеспечивает достаточное быстродействие, то при запуске второй базы общая производительность снижается в два раза, даже если вторую базу использует только один пользователь — кэш в этом случае делится пополам.

3.3. Разработка экранных форм

Комплекс складского учета, созданный с целью ускорить работу персонала, должен обладать интуитивно понятный для базового пользователя

интерфейс, и позволять получить быстрый доступ к своим функциям. Результатом внедрения подобной программы будет ускорение работы персонала и повышение эффективности персонала. Использовать данное ПО может любой пользователь, как опытный, так и имеющий базовый уровень компьютерной грамоты.

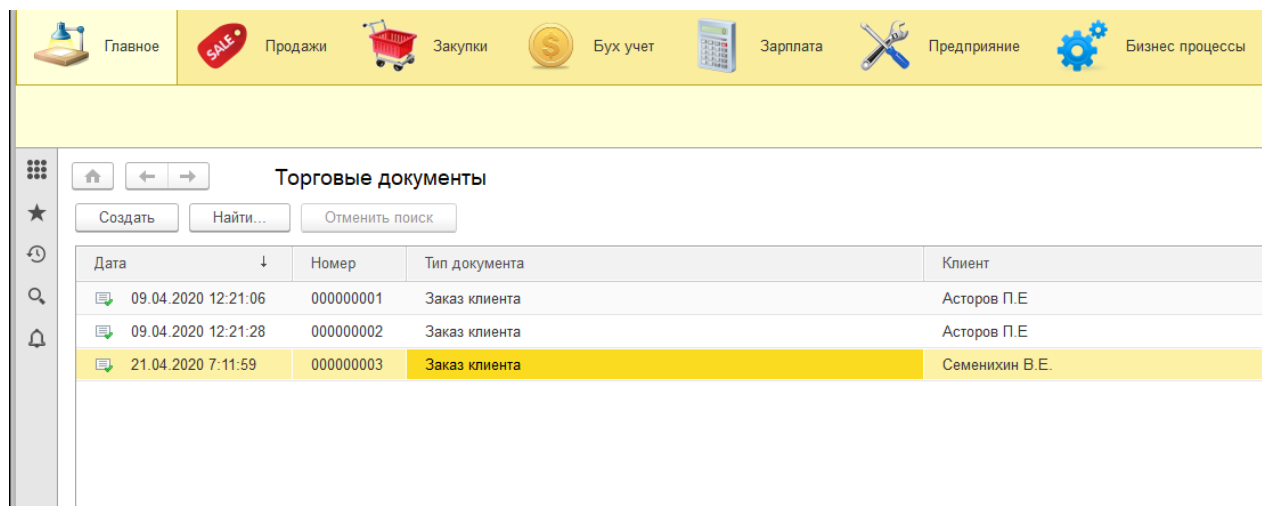


Рисунок 3.3.1 – Окно «Торговые документы»

При открытии программного комплекса нас встречает выбор из позиций меню: «Продажи», «Закупки», «Бух Учет», «Зарплата», «Предприятие», «Бизнес Процессы».

Начнем с модуля «Продажи». Модуль состоит из следующих разделов:

Выписка банка – Раздел созданный для ведения, редактирования и полного контроля над расходными взаимодействиями с банком;

Заказ клиента – Раздел для работы с накладной заказами клиента, с указанием всех требуемых полей (данные о клиенте, сотрудник проводящем операцию, складе, статусе сделки, и также имеет идентификатор заказа).

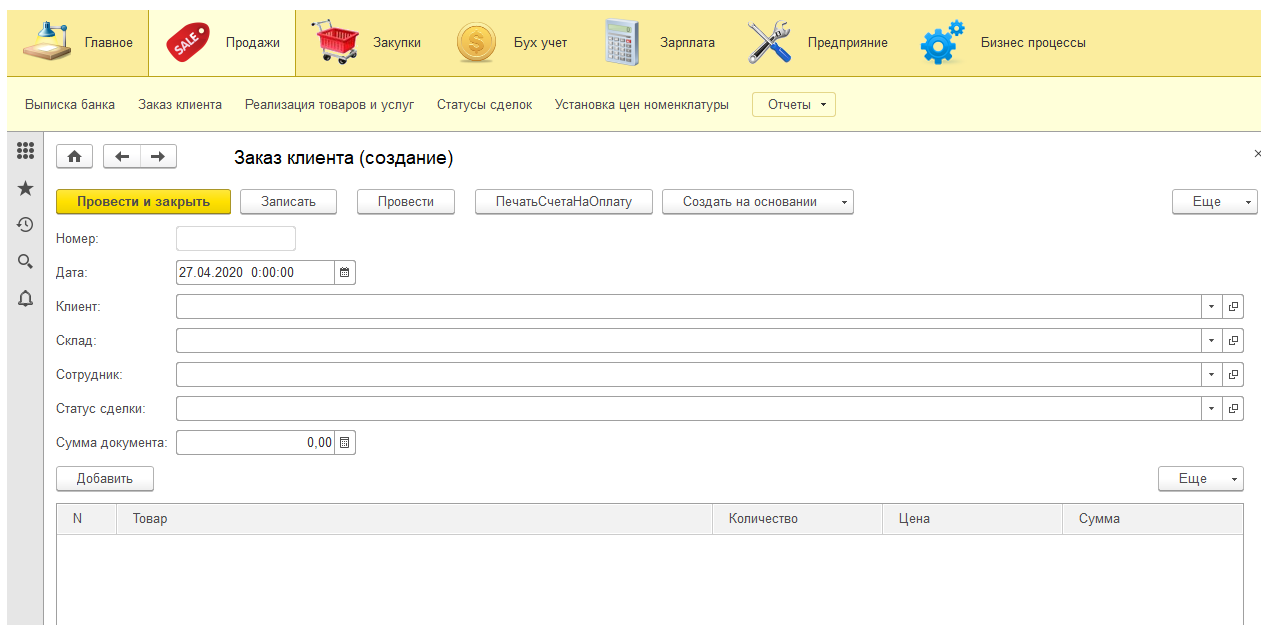


Рисунок 3.3.2 – Окно «Заказ клиента»

После внесения требуемых данных заказа переходим к следующим полям. А именно статус сделки. Статус сделки описывает нынешнее состояние заказа. Имея следующие состояния: «Осуществлен», «В рассмотрении», «Заморожен». В данном модуле находится внесение товаров в заказ и внесения всех требуемых данных о товаре (количество, стоимость за единицу товара, и т.д.).

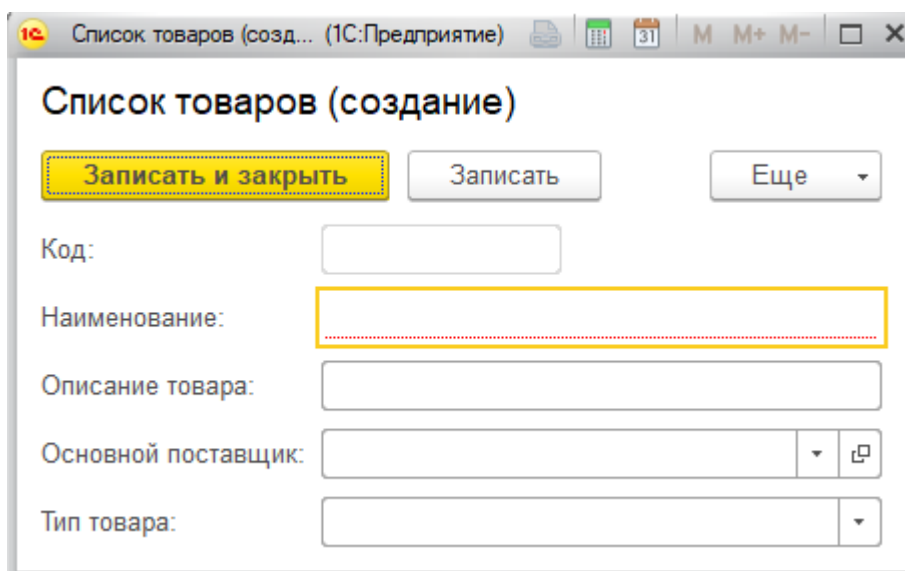


Рисунок 3.3.3 – Форма «Список товаров»

Пройдемся по позициям вносимой информации.

Код - автоматически вносимое поле-идентификатор. Предназначено для идентификации конкретного заказа, даже при абсолютном сходстве в остальных полях, заказы будут отличаться полем-идентификатором.

Наименование товара и Описание товара требуются для того, чтоб дать понять заказчику, что из себя представляет товар, и содержит краткую характеристику товара.

Основной Поставщик – поле содержащее указание на предприятие, осуществляющее поставку данного товара на склад.

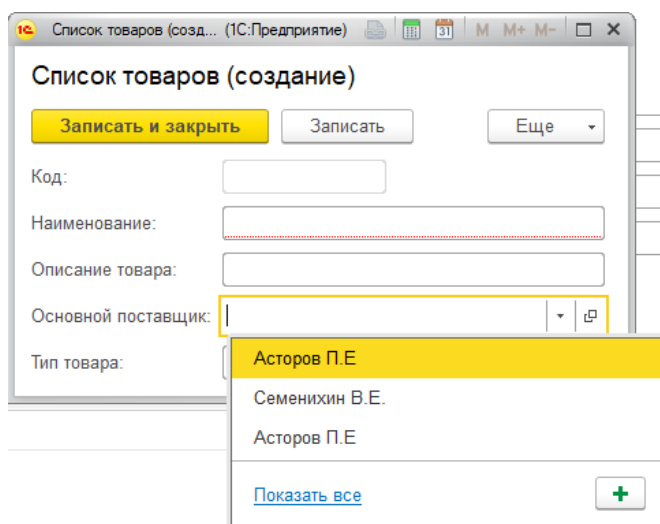


Рисунок 3.3.4 – Форма «Список товаров»

Выбор осуществляется из списка поставщиков, работающих с данным предприятием. В случае, если данный поставщик является новым партнером фирмы, управляющий может добавить поставщика нажав на кнопку добавления нового поставщика, окрашенную в зеленый цвет.

Следующее поле «Тип Товара» определяет группу товара.

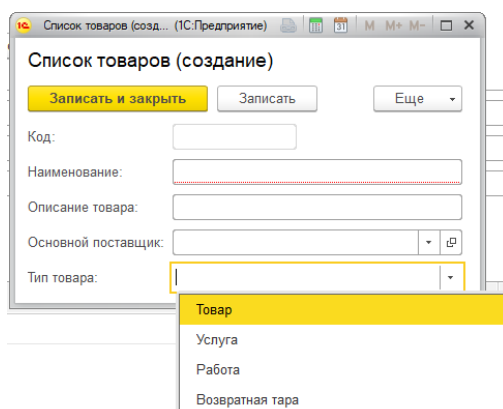


Рисунок 3.3.5 – Форма «Список товаров»

Выбор стоит между следующими подкатегориями товара. «Товар» представляет из себя материально представленный товар выставляемый на продажу поставщиком. «Услуга» представляет собой какой-то комплекс действий осуществляемый подрядной организацией. Последующие подкатегории «Работа» и «Возвратная Тара» также важны, но используются реже.

Реализация Товаров и Услуг - модуль созданный для ведения всей требуемой документации в данном направлении торговли, и имеющий все необходимые уточнения. В конкретную запись пользователь может добавить большое количество характеристик, что несомненно качественно выделяет данную программу на фоне конкурентов.

Реализация товаров и услуг (создание)

Основное [Взаиморасчеты](#) [Продажи](#) [Себестоимость товаров](#) [Товары на складах](#)

Провести и закрыть Записать Провести

Номер:

Дата:

Клиент:

Склад:

Сотрудник:

N	Товар	Количес
---	-------	---------

Рисунок 3.3.6 – Форма «Реализация товаров и услуг»

Статусы сделок – модуль для создания категорий статусов. Имея при себе категории: В рассмотрении, Осуществлен и Заморожен. При желании пользователь может создать и добавить собственную категорию, в зависимости от направления работы предприятия. Также для краткого указания категории могут быть дополнительно наименованы порядковыми номерами.

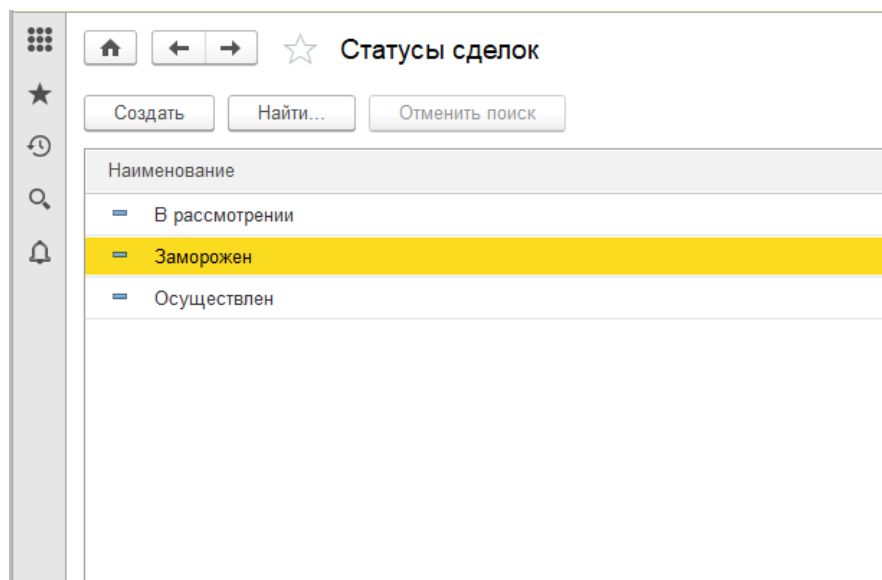


Рисунок 3.3.7 – Окно «Статусы сделок»

И последним мини-модулем является «Установка Цен на Номенклатуры».

Осуществляет работу в направлении ценовой политики относительно номенклатур.

Важным является то, что в данном модуле программы возможно осуществить работу с отчетами. Модуль предоставляет и обеспечивает работающего широким списком обобщающих и аналитических отчетов, позволяющих быстро и просто найти любую интересующую информацию в простой для понимания экранной форме.

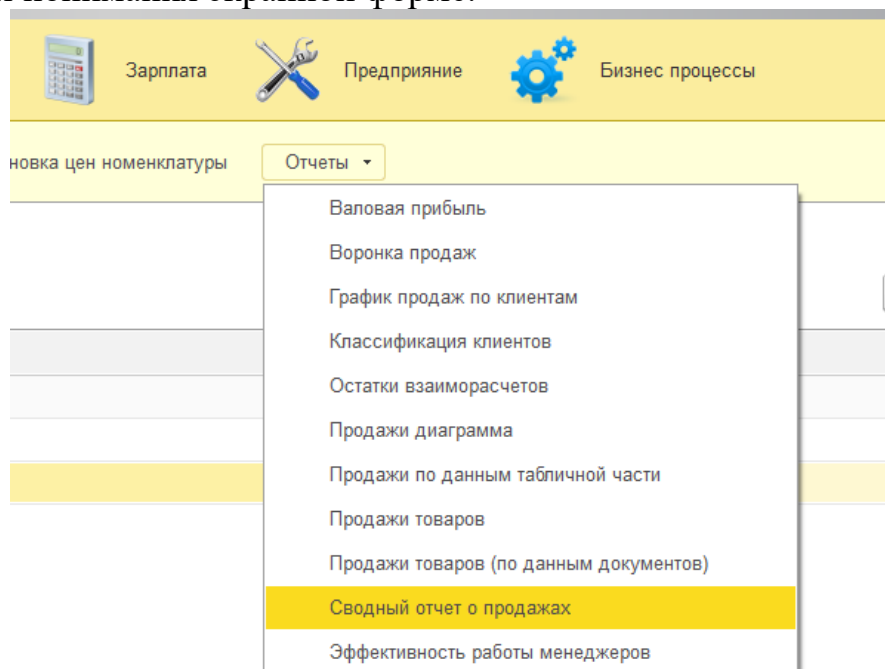


Рисунок 3.3.8 – Содержимое вкладки «Отчеты»

На данном изображении указан подробный набор диаграмм и технических документов. Диаграммы могут быть использованы для подробного изучения процессов, проходящих внутри предприятия. Для отдела бухгалтерии - это невероятно важный набор расчетов, требуемый для выполнения операций осуществляемых бухгалтерским персоналом.

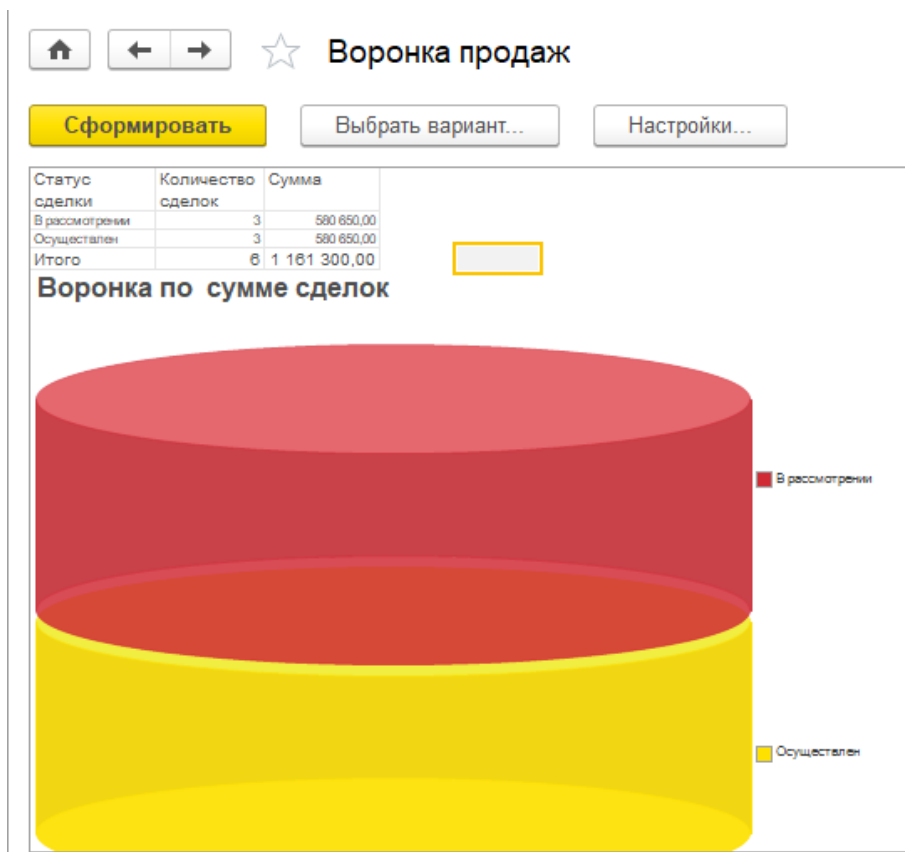


Рисунок 3.3.9 – Пример воронки продаж

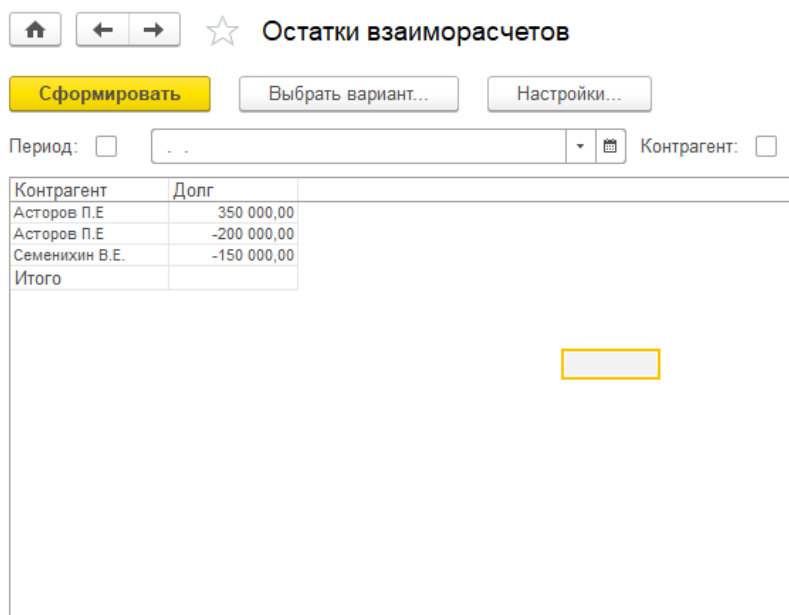


Рисунок 3.3.10 – Пример статистики «Остатки взаиморасчетов»

С данными расчетами можно проводить большое количество действий. Как например важную функцию выводу документа на печать напрямую из программы.

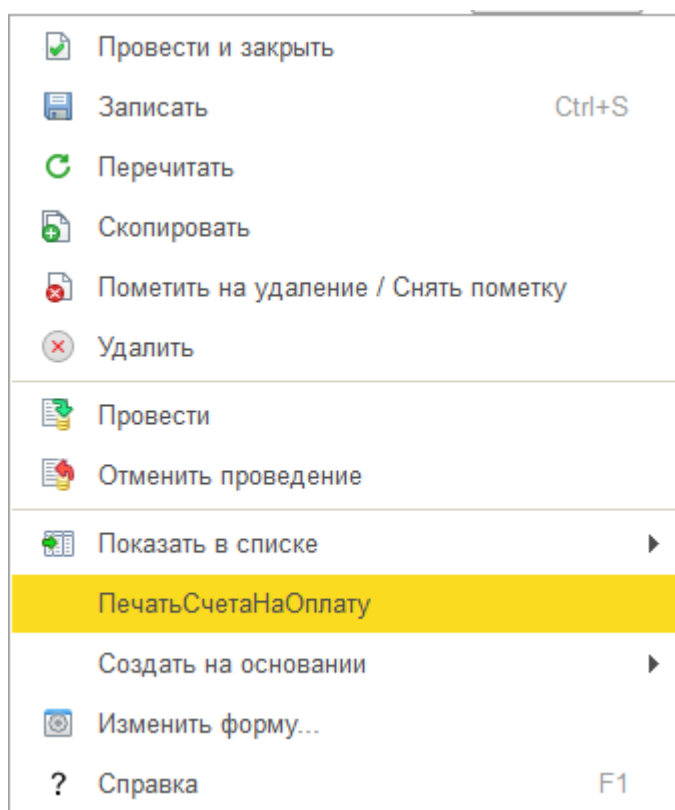


Рисунок 3.3.11 – Вывод на печать

Для рассмотрения данной функции рассмотрим контекстное меню модуля создания Заказа. После создания и полной корректировки и заполнения заказа есть возможность вывести данный заказ в виде счета на оплату.

Счет на оплату

Номер 000000003
 Дата 21.04.2020 7:11:59
 Клиент Семенихин В.Е.

№	Товар	Количество	Цена	Сумма
1	Испанский хамон	30,000	9 000,00	270 000,00
2	Картофель	120,000	120,00	14 400,00
3	Албанский изюм	15,000	2 200,00	33 000,00

Бубликов В.В. _____

Склад Склад на Шолохова

Рисунок 3.3.12 – Счет на оплату

Итак, перед вами счет на оплату для нашего предприятия, осуществляющего реализацию товаров бытового и ресторанного использования. Счет содержит необходимую информацию для пользователя, дату его осуществления, данные клиента, наименования товара отпускаемого по данному счету, цены товара, его количество и итоговая сумма заказа, подлежащая оплате.

И один из важнейших разделов для работы предприятия. Раздел «Товары»

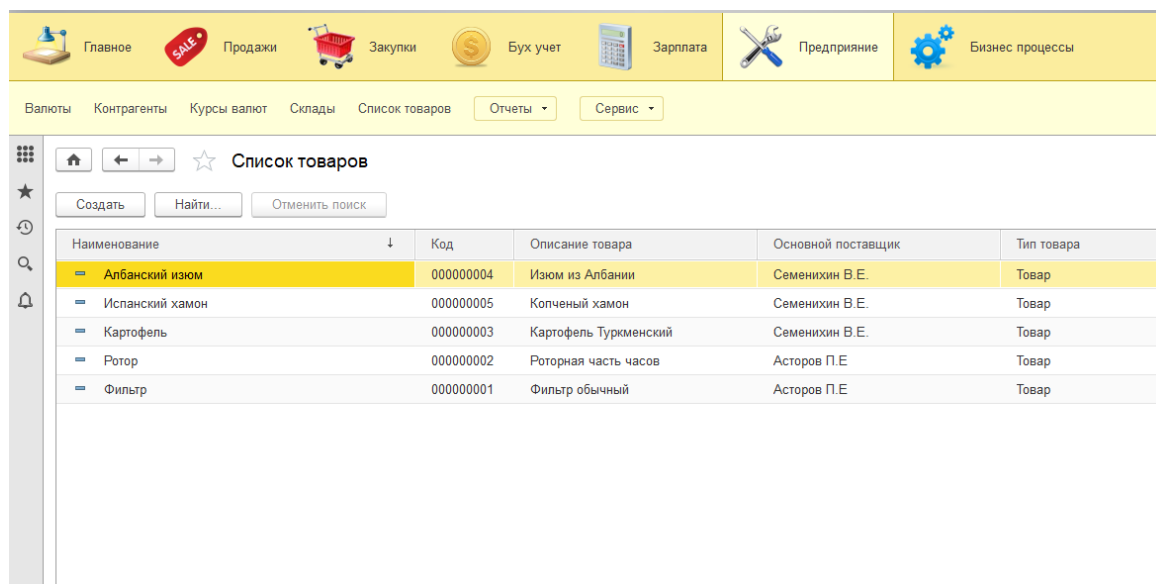


Рисунок 3.3.13 – Окно «Список товаров»

Раздел со всеми имеющимися товарами предприятия, подлежащими реализации и доступные для покупки клиентом.

4 Техничко-экономическое обоснование

Темой дипломного проекта является «Разработка системы учета движения товаров на складе малого предприятия»»

В данной дипломной работе описан проект по разработке системы учета движения товаров на складе малого предприятия на платформе 1С, разработана конфигурация для предприятия.

Целью данного проекта является – обеспечение предприятия системой учета движения товаров и комплексом учета движения товаров, включающим в себя ведение первичной отчетности, внутренней и бухгалтерской документации. Данная конфигурация облегчает труд сотрудников компании, перекладывая многие задачи на компьютер, а также имеет в себе ведение первичной отчетности, внутренней и бухгалтерской документации, ранее не представленных в конфигурации 1С.

В разработке программного продукта примет участие качественно подобранная команда IT-специалистов, состоящая из таких работников, как: технический директор, develop-программист. Технический директор осуществляет и следит за графиками работы, контролем персонала и его полезными действиями, в первую очередь за качество будет отвечать именно технический руководитель. Develop-программист в свою очередь занят разработкой технической части проекта, разработкой программной части приложения, всеми сопутствующим этому тестами и готовит приложение с самого начального и до конечного выпуска продукта в свет. Соответственно, весь основной процесс разработки осуществляет программист, на плечи которого и ложится большая часть задач, пока как технический директор совершает дела вне программного осуществления, а именно сопутствующая этому организационная работа. Техничко-экономическое обоснование имеет следующие пункты:

- определение сложности разработки программного обеспечения;
- расчет затрат на разработку ПО;
- определение ценности готового продукта;
- оценка результатов работы программного обеспечения.

1.1 Определение сложности разработки ПО

Для того, чтобы точно определить сложность разработки программного обеспечения, необходимо произвести деление всей задачи на более простые этапы. Это позволит эффективно следить за прогрессом разработки программного обеспечения, за счет деления сложной задачи на более легкие подзадачи. Такой подход, с моей точки зрения, считается более эффективным и позволяет результативно и быстро обрабатывать подзадачи. Модель

распределения сложности разработки ПО и стадии разработки представлены в таблице 4.1.

Таблица 4.1 – Этапы разработки ПО

Этапы разработки ПО	Вид работы	Трудоемкость, чел. час.
Этап 1	Постановка задач	10
Этап 2	Разработка и утверждение ТЗ на разработку ПО	20
Этап 3	Поиск и изучение подобных программ	15
Этап 4	Поиск и изучение сопутствующей литературы	10
Этап 5	Составление аналитических графиков ПО	5
Этап 6	Оформление теоретической части дипломной работы	15
Этап 7	Разработка практической части дипломного проекта	25
Этап 8	Реализация проекта	50
Этап 9	Отладка и устранение недоработок	20
Этап 10	Организация отчета и результатов работы	10
Этап 11	Тестирование ПО	20
Этап 12	Подведение итогов по разработанному ПО	10
Этап 13	Внедрение	15
Итого: трудоемкость выполнения дипломного проекта		225

Продолжительность рабочего дня равна 8 часам. В результате для реализации программного обеспечения необходимо 28 рабочих дней.

1.2 Расчет затрат на разработку ПО

Определение затрат необходимых для разработки программного обеспечения производится на основе имеющейся сметы, которая включает следующие элементы:

- материальные затраты;
- затраты на оплату труда;
- социальный налог;
- амортизация основных фондов;
- прочие затраты.

Материальные затраты делятся на основные и вспомогательные затраты на материалы, энергию и другие затраты необходимые для разработки ПО. Расчет материальных затрат происходит по форме, предоставленной в таблице 4.2.

Таблица 4.2 – Затраты на материальные ресурсы

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Бумага для офиса	Svetocopy	Пачка	3	1000	3000
Тетрадь (96 листов)	ErichCrause	Штук	2	190	380
Блокнот	ErichCrause	Штук	2	400	800
Ручки	Bruno Visconti	Штук	2	150	300
Компьютерная мышь	Logitech	Штук	1	3000	3000
Итого:					7 480

Для разработки программного обеспечения будет использоваться ноутбук Asus Vivo X540LJ, мощности ноутбука достаточно для выполнения поставленных задач. Ввиду того, что ноутбук содержит установленную операционную систему и программное обеспечение необходимое для разработки ПО, нет нужды производить дополнительные расходы на новые ОС и ПО.

Общую сумму, необходимую на материальные средства (Z_m) можно рассчитать по следующей формуле:

$$Z_m = \sum P_i * C_i, \quad (4.1)$$

где P_i - расход i -го вида материального ресурса, натуральные единицы;

C_i - цена за единицу i -го вида материального ресурса, тг;

i - вид материального ресурса;

n - количество видов материальных ресурсов.

Расчет затрат на необходимое оборудование и программное обеспечение производится по форме, приведенной в таблице 4.3.

Таблица 4.3 – Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	Asus Vivo X540LJ	Штук	1	250000	250000
Принтер	Samsung SCX-3400	Штук	1	47000	47000
Модем	TP-LINK TD	Штук	1	14000	14000
ОС	Windows 10	Штук	1	-	-
Итого:					311000

$$Z_m = 7480 + 311000 = 318480 \text{ (тг)}$$

Для реализации программного обеспечения необходимы материалы на сумму 318 480 тенге.

4.3 Расчет затрат на электроэнергию

Так как при разработке программного обеспечения не обойтись без потребления электроэнергии, имеет смысл произвести расчет затрат на электроэнергию.

Согласно таблице 4.1 для разработке программного обеспечения необходимо порядка 225 часов, теперь необходимо рассчитать стоимость электроэнергии, которая будет потрачена в течении 225 часов. Для принтера расчет будет проводиться для периода в 24 часа, так как нет необходимости постоянно использовать принтер.

$$Z = Z_{\text{эл.эн.обор.}} + Z_{\text{доп.нужды.}} \quad (4.2)$$

где $Z_{\text{эл.эн.обор.}}$ – затраты на электроэнергию оборудования;

$Z_{\text{доп.нужды.}}$ – затраты электроэнергии на дополнительные нужды.

Расчет электроэнергии, которая необходима для оборудования определяется по следующей формуле:

$$Z_{\text{эл.эн.обор.}} = \sum W * K_{\text{исц}} * S * T, \quad (4.3)$$

где W – потребляемая мощность, Вт;

$K_{\text{исц}}$ – коэффициент использования ($K_{\text{исц}} = 0,7..0,9$);

T – время работы;

S – тариф (1кВт/ч = 19,17 тенге).

Итоги по расчетам стоимости затрачиваемой электроэнергии представлены в таблице 4.4.

Таблица 4.4 – Затраты на электроэнергию

Наименование приборов	Паспортная мощность, кВт	Коэффициент мощности	Время работы оборудования, ч	Цена ЭЭ /кВтч	Сумма, тг.
Ноутбук	0,6	0,7	225	19,17	1811,5
Модем	0,08	0,9	225	19,17	310,5
Принтер	0,5	0,9	24	19,17	207
Кондиционер	0,8	0,9	180	19,17	2484,4
Освещение	0,3	0,7	225	19,17	905,7
Итого:					5719

$$Z_{\text{эл.эн.обор.}} = 5719 \text{ (тенге)}$$

На дополнительные потребности расходы подсчитываются на основе повышенного показателя в объеме 5% от расходов на электроэнергию:

$$Z_{\text{доп.нужды}} = 5\% * Z_{\text{эл.эн.обор.}} \quad (4.4)$$

Определим затраты на дополнительные потребности согласно формуле (4.4):

$$Z_{\text{доп.нужды}} = 0.05 * 5719 = 285,95 \text{ (тенге)}$$

Исходя из всех расчетов, полные расходы на электроэнергию составляют:

$$Z = 285,95 + 7117 = 6004,95 \text{ (тенге)}$$

4.4 Расчет затрат на оплату труда

Для разработки программного обеспечения, как указывалось ранее, необходимо два работника:

- руководитель проекта – управление рабочим временем, корректировка рабочих процессов, координация, изучение предметной области;
- разработчик – разработка ПО, тестирование и сопровождение.

Сумму расходов на оплату труда можно рассчитать по следующей формуле:

$$Z_{\text{тр}} = \sum ЧС_i * T_i \quad (4.5)$$

где $ЧС_i$ - часовая ставка i -го работника, тг;

T_i - трудоемкость разработки модели, чел.×ч; i - категория работника;

n - количество работников, занятых разработкой ПП.

Во время реализации проекта рабочее время участников не равномерно, поэтому имеет смысл установить часовую ставку каждого работника и общий объем заработной платы.

Часовую ставку сотрудника можно рассчитать по следующей формуле:

$$ЧС_i = \frac{ЗП_i}{ФРВ_i} \quad (4.6)$$

где $ЗП_i$ - месячная заработная плата i -го работника, тг;

$ФРВ_i$ - месячный фонд рабочего времени i -го работника, час.

Месячная заработная плата руководителя равняется 190 000 тенге и месячная заработная плата разработчика равняется 160 000 тенге. Рассчитаем часовую ставку каждого работника согласно формуле (4.6):

$$ЧС_{\text{руководитель}} = \frac{190\,000}{22 * 8} = 1\,079,54 \text{ тг/ч}$$

$$ЧС_{\text{разработчик}} = \frac{160\,000}{22 * 8} = 909,09 \text{ тг/ч}$$

Часовая ставка руководителя составляет 1 079,54 (тг/ч), трудоемкость разработки равняется 100 часам. Часовая ставка разработчика составляет 909,09 (тг/ч), трудоемкость разработки равняется 225 часам. Согласно

формуле (4.5) можно рассчитать сумму расходов на заработную плату работников:

$$Z_{\text{тр}} = 1079,54 * 100 + 909,09 * 225 = 107954 + 204545,25 = 312499,25$$

Расчеты затрат по оплате труда показаны в таблице (4.5).

Таблица 4.5. – Расчет заработной платы

Категория работника	Квалификация	Трудоемкость разработки ПП, час.	Часовая ставка, тг/ч	Сумма, тг.
Руководитель	Проектный руководитель	100	1079,54	107954
Разработчик	Программист	225	909,09	204545,25
Итого:				312500

4.5 Расчет затрат по социальному налогу

Согласно Налоговому кодексу Республики Казахстан социальный налог составляет 9,5% от фонда оплаты труда. Налоги уплаченные за работников юридическим лицом можно рассчитать по следующей формуле:

Уплаченные юридическим лицом налоги	10,46	ФОТ	312 500
СО (Социальные отчисления)	3,5	(ЗП - ОПВ)*3,5%	9 843,75
ВОСМСЮ (Отчисления на ВОСМСЮ)	2,0	ЗП*2%	6 250,00
СН (Социальный налог)	9,5	(ЗП - ОПВ - ВОСМСФ)*9,5%-СО	16281,25
Всего уплаченные налоги			32 375

Результаты расчетов представлены в таблице (4,6):

Таблица 4.6 – Начисление социального налога

Категория работника	Количество человек	Заработная плата, тг	Пенсионные отчисления, тг	Социальный налог, тг
Руководитель	1	107954	10795	11184
Разработчик	1	204545,25	20454	21190,88
Итого:				32 375

4.6 Амортизация основных фондов и прочие затраты

Нормы амортизации ОФ необходимо определить в соответствии с налоговым кодексом РК. Амортизацию ОФ можно определить по следующей формуле:

$$A_r = \frac{C_{об} * H_a}{100} \quad (4.8)$$

где, $C_{об}$ – стоимость оборудования;

H_a – норма амортизации (норма амортизация = 25);

Формула (4.8) позволяет рассчитать нужную сумму для амортизационных отчислений за год для ноутбука:

$$A_r = \frac{250\,000 * 25}{100} = 62\,500 \text{ тенге}$$

Теперь необходимо рассчитать норму амортизации за период разработки:

$$A_r = \frac{62500 * 34}{365} = 5821,9 \text{ тенге}$$

Подобным образом необходимо рассчитать норму амортизации для всего оборудования. Результаты расчетов приведены в таблице (4.7).

Таблица 4.7 – Амортизация ОФ

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Сумма амортизации за год, тг	Сумма амортизации за время разработки, тг
Ноутбук	250 000	25	62500	5821,9
Принтер	47000	25	11750	96,57
Модем	14 000	20	2 800	172,6
Итого:			77050	6091,07

Смета расходов на разработку ПО.

На основе всех представленных расчетов необходимо оформить смету расходов на разработку ПО согласно форме, которая приведена в таблице (4.8).

Таблица 4.8 – Смета затрат на разработку ПО

Статьи затрат	Сумма, тг
Затраты на оборудование	318480
Затраты на программное обеспечение	0
Затраты на оплату труда	312499,25
Социальные налоги	32 375
Затраты на электроэнергию	6004,95
Амортизация основных фондов	6091,07
Прочие расходы	27 600
Итого по смете:	703331,37

На рисунке (4.6.1) продемонстрирована диаграмма рабочих расходов.



Рисунок 4.6.1 – Диаграмма затрат

4.7 Определение возможной (договорной) цены ПО

Стоимость программного обеспечения определяется на основе качества разработанного продукта, сроков его разработки и производительности продукта. Стоимость Ц_д для программного обеспечения можно рассчитать по следующей формуле:

$$Ц_{д} = Z_{\text{нир}} \left(1 + \frac{P}{100} \right), \quad (4.9)$$

где $Z_{\text{нир}}$ – затраты на разработку программного обеспечения, тг;

P – средний уровень рентабельности ПО, (%). Данный параметр принят равным 25%.

$$C_d = 703331,37 \left(1 + \frac{25}{100} \right) = 879164,21 \text{ тенге}$$

Далее необходимо определить стоимость реализации с учетом НДС, ставка НДС устанавливается законодательством РК. На 2019 года ставка НДС составляет 12%. Стоимость реализации учитывая НДС можно рассчитать по следующей формуле:

$$C_p = C_d + C_d * \text{НДС}, \quad (4.10)$$

$$C_p = 879164,21 + 879164,21 * 0,12 = 984663,91 \text{ тенге}$$

Данную цену можно округлить до 984664 тенге.

4.8 Сравнительный анализ эксплуатационных затрат до и после внедрения

Рассмотрим список расходов после установки программного обеспечения:

- основная заработная плата проектного руководителя и developer-программиста проекта включая отчисляемыми налогами;
- материальные ресурсы;
- амортизация основных фондов;
- накладные расходы.

Список расходов до внедрения:

- основная заработная плата заведующего складом, бухгалтера и 2 работников отдела закупок;
- материальные ресурсы;
- накладные расходы.

Расчет затрат до внедрения ПО

Таблица заработной платы и отчисляемых налогов представлена в таблице 4.8.1.

Таблица 4.8.1- Затраты на заработную плату

Сотрудник	Кол-во, чел	Заработная плата в месяц, тг	Заработная плата в год, тг	Социальное отчисление	Социальный налог	ВОСМС
Заведующий складом	1	190000	2 280 000	71820	118788	45600
Бухгалтер	1	160000	1 920 000	60480	100032	38400
Закупщик	2	100000 (200 000)	1 200 000 (2 400 000)	37800 (75600)	62520 (125040)	24000 (48000)
К оплате						7283760

Расчет затрат на необходимое для продукта ПО и оборудование представлен в таблице 4.8.2

Таблица 4.8.2 – Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	Lenovo IdeaPad S145	Штук	3	185000	555000
Принтер	Samsung SL-M2020W A4-W	Штук	1	43935	43935
Компьютерная мышь беспроводная	Logitech M90 Grey, 910-001794	Штук	3	2990	8970
Итого					607905

Затраты, отнесенные к амортизационным отчислениям представлены в таблице 4.8.3

Таблица 4.8.3-Амортизационные отчисления

Оборудование	Количество	Стоимость, тенге	Отчисления
Lenovo IdeaPad S145	3	555000	12924.65
Samsung SL-M2020W A4-W	1	43935	1083.32
Итого	4	598935	14 007.9

Общие накладные расходы вычисляются по формуле:

$$З_{н.р.} = 7283760 * 0,2 = 1\,456\,752 \text{ тенге.}$$

В итоге, общие эксплуатационные расходы до внедрения программного обеспечения будут равны,

$$З = 7283762 + 607905 + 14\,007.9 + 1\,456\,752 = 9362426 \text{ тенге.}$$

Расчет затрат после внедрения ПО

Таблица заработной платы и отчисляемых налогов представлена в таблице 4.8.4

Таблица 4.8.4 - Затраты на заработную плату

Сотрудник	Кол-во, чел	Зарботная плата в месяц одного, тг	Зарботная плата в год, тг	Социальное отчисление	Социальный налог	ВОСМС
Руководитель	1	107954	1295448	40806,6	134208	25908,96
Разработчик	1	204545,25	2454543	77318,1	254280	49090,8
К оплате						4331602,92

Расчет затрат на необходимое для программного продукта и оборудование представлен в таблице 4.8.5

Таблица 4.8.5 – Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	Lenovo IdeaPad S145	Штук	1	185000	185000
Принтер	Samsung SL-M2020W A4-W	Штук	1	43935	43935
Компьютерная мышь беспроводная	Logitech M90 Grey, 910-001794	Штук	1	2990	2990
Итого					231925

Затраты, отнесенные к амортизационным отчислениям представлены в таблице 4.8.6

Таблица 4.8.6 - Амортизационные отчисления

Оборудование	Количество	Стоимость, тенге	Отчисления
Lenovo IdeaPad S145	1	185000	4308.21
Samsung SL-M2020W A4-W	1	43935	1083.32
Итого	2	228935	5391.5

Общие накладные расходы рассчитываются по формуле:

$$З_{н.р.} = 4331602,92 * 0,2 = 866320.584 \text{ тенге.}$$

Итого, общие эксплуатационные расходы после внедрения программного обеспечения будут равны,

$$З = 4331602.92 + 231925 + 5391.5 + 866320.584 = 5435240 \text{ тенге.}$$

Для большей ясности предоставим все данные в сравнительную таблицу
4.8.7

Таблица 4.8.7 - Годовые эксплуатационные затраты

Статьи	До внедрения ПП	После внедрения ПП
Годовая заработная плата	7283760	5435240
Оборудование	607905	231925
Амортизационные отчисления	14 007.9	5485,65
Накладные расходы	1456752	866320.584
Всего	9362424.9	6538971.234

Годовая экономия денежных средств будет равна согласно расчету:
 $9362424.9 - 6538971.234 = 2823453$ тенге.

Также определим суммарные выгоды от внедрения программного продукта.

Технико-экономические показатели позволяют рационально определить целесообразность разработки ПО и ее установки, а также взвесить реальную выгоду компании, как для разработчика системы, так и для ее пользователя.

Также определить срок окупаемости программного обеспечения в месяцах можно по формуле:

$$P_{\text{окуп}} = 12 * \frac{Z_{\text{разработка}}}{\Delta_{\text{годовая}}} \quad (4.21)$$

где $P_{\text{окуп}}$ – период окупаемости в месяцах;

$Z_{\text{разр.}}$ – затраты на разработку программы;

$\Delta_{\text{годовая}}$ – годовая экономия.

$$P_{\text{окуп.}} = 12 \cdot \frac{984664}{2823453} = 12 \cdot 0,3487 = 4,1 \text{ мес.}$$

Предвидя годовую выгоду продукта и годовые эксплуатационные затраты с внедрением программного обеспечения можно определить коэффициент эффективности ($K_{\text{эф.}}$) по формуле:

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_p = \frac{\Delta_{\text{уг}}}{K}$$

$$E_p = \frac{2823453}{984664} * 100\% = 286\% \text{ или } 2,86 \text{ тенге.}$$

Коэффициент эффективности указывает на соотношение сэкономленных тенге и потраченных. В заключении можно с уверенностью заявить о том, что на 1 затраченный тенге приходится 2,86 тенге экономии.

Внедрение данного программного продукта в рабочий процесс предприятия типа Ногеса приносит экономию средств и автоматизацию рабочего процесса. Установка данного программного комплекса позволяет сократить штат сотрудников, что в следствии сокращает затраты на выплату налогов, зарплат и прочих денежных расходов. Соответственно, уменьшится объем технического оснащения, амортизационные отчисления и затраты на электроэнергию. Программный комплекс складского выгодно автоматизирует процесс, облегчив труд персонала и увеличив его КПД, к тому же уменьшив расходы предприятия.

4.9 Вывод по экономической части

Данная глава дипломного проекта содержит экономические расчеты, которые позволяют определить затраты необходимые для разработки программного продукта. Расчеты включают в себя:

- расчет трудоемкости разработки программного продукта;
- расчет затрат на разработку программного продукта;
- расчет затрат на электроэнергию;
- расчет затрат на оплату труда;
- расчет затрат по социальному налогу;
- амортизация основных фондов и прочие затраты.

Для покупателя основным показателем будет считаться оптимальная цена программного продукта и его производительность. Цена и полезность программного продукта должна обладать равновесием, чтобы покупатель был заинтересован в приобретении разработки. Качественным результатом для покупателя считается, что приобретенное программное обеспечение полностью покрывает все необходимые задачи, которые встают перед покупателем. Так же в последней главе был произведен расчет договорной цены программного продукта, который равняется 984664 тенге, данное значение является рациональным с точки зрения экономической эффективности.

5. Безопасность жизнедеятельности

В данной дипломной работе описан проект по разработке системы учета движения товаров на складе малого предприятия на платформе 1С, разработана конфигурация для предприятия.

Помещением является пространство для технической оснастки склада предприятия и пространство работы менеджеров и складского отдела учета. Программный комплекс для ведения складского учета был разработан для компании сегмента Нoreca. С целью обеспечения персонала безопасным местом работы был проведен комплексный анализ рабочего места на риски и организована рабочая площадка с комфортными условиями труда.

5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал

Рабочее место персонала должно отвечать техническим нормам, предназначенным для данного помещения, с учетом наличия всех защитных правил, процедур и критериев во избежание воздействия на здоровье персонала производственных факторов способных нанести вред как в краткосрочной, так и в долгосрочной перспективе.

Показатели данных факторов должны находиться в интервале, не превышая допустимые значения предусмотренные правовыми, техническими и санитарно-техническими нормами. Нормативные правовые акты в области безопасности и охраны труда устанавливают организационные, технические, технологические, санитарно-эпидемиологические, медицинские и иные нормы, правила, процедуры и критерии, направленные на сохранение жизни и здоровья работников в процессе их трудовой деятельности.

Наиболее характерными факторами риска для работников офиса (согласно закону N 528-2 «о безопасности и охране труда») являются перенапряжение зрения и внимания; превышение норм электромагнитного, рентгеновского, ультрафиолетового и инфракрасного излучения; чрезвычайно высокий уровень статического электричества и показатель запыленности воздуха рабочей зоны, длительное нахождение в сидячем положении, монотонность труда, повышенное содержание в воздухе рабочей зоны двуокиси углерода, озона, аммиака, фенола и формальдегида. Часто работники офисных помещений жалуются на следующие симптомы: жжение в глазах, покраснение век и конъюнктивы, чувство инородного тела или песка под веками, боли в области глазниц и лба, затуманивание зрения, замедленная перефокусировка с ближних объектов на дальние. Причиной тому является переутомление зрительного аппарата при длительной работе с экраном без перерывов. Данные симптомы имеют наименование данное Всемирной Организацией Здравоохранения, и именуется как «компьютерный зрительный

синдром».

Во избежание появления данных симптомов у работников, должны быть приняты следующие меры относительно рабочего пространства и экрана монитора [24]:

- высота рабочей поверхности стола должна составлять 680-800мм;
- покрытие стола должно быть диффузно отражающим (не глянцевым) с коэффициентом отражения 0,45-0,50;
- расположение экрана должно быть таким, чтобы изображение в любой его части было различимо без необходимости подъема или опускания головы;
- расположение монитора должно быть ниже уровня глаз, угол наблюдения не должен превышать 60°
- яркость бликов на экране дисплея не должна превышать 40 кд/м²;
- неравномерность яркости рабочего поля не должна превышать 20%;
- неравномерность яркости элементов знака не должна превышать 20%;
- яркостный контраст изображения должен быть не менее 3:1;

В качестве решения требуется обеспечить работника рабочим местом соответствующим вышеперечисленным требованиям и соблюдение работником верного ритма работы с экраном монитора, а именно: наличие общей длительности перерывов от 50 минут до 1,5 часа при 8-часовом рабочем дне, проведение зрительной гимнастики, и прием необходимых микроэлементов [25].

Система складского учета будет опробована в пространстве работы менеджеров и отдела учета при складе малого предприятия.

На нижеприведенном рисунке показан план данного помещения. Данное рабочее пространство предназначено для 7 работников предприятия, а именно: директор отдела учета, 3 менеджера закупа, 3 менеджера отдела учета.

Персоналу требуется рабочее место обеспечивающее комфортные условия для работы. Рабочее пространство не должно повышать уровень утомления тем самым снижая работоспособность персонала.

Рабочее помещение имеет следующие характеристики: длина – 10 м, ширина – 6 м, высота – 4 м, высота окошка – 2 м. Пространство обладает как искусственным, так и естественным освещением. Естественное освещение доступно благодаря наличию двух окон с габаритами 4х2 метра. Каждое отдельное рабочее место оснащено моноблоком для рабочего использования. Общее число моноблоков равно 7 единицам.

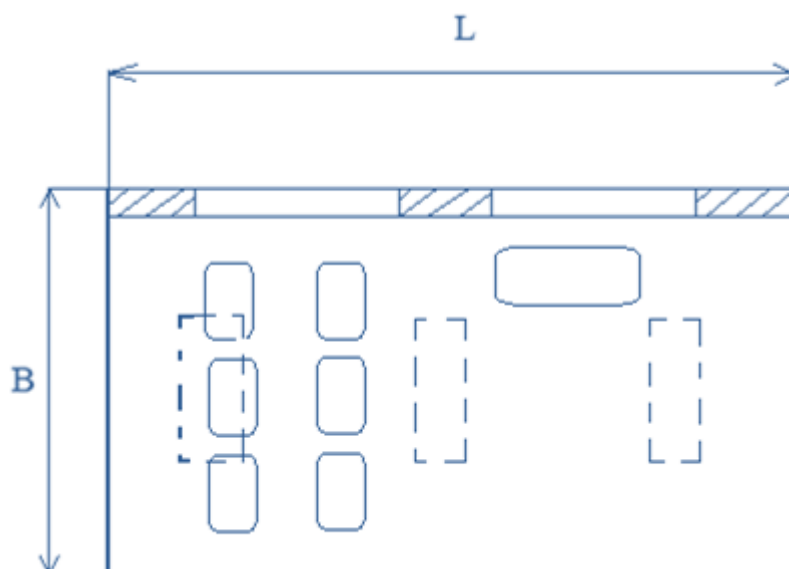


Рисунок 5.1.1 – План учебного помещения

Организация рабочего места является процессом определения задач, которые должны выполняться работником, носителем конкретных знаний, навыков и умений, а также соответствовать ряду требований, прежде всего в области санитарии, техники безопасности, правил эксплуатации оборудования и т. п.

5.2 Расчет естественного освещения

Первым делом осуществляем проверку площади боковых световых проветров здания, предназначенных для обеспечения необходимыми показателями освещенности рабочего пространства.

Характеристики помещения: длина $L = 10$ м, ширина $B = 6$ м, высота $H = 4$ м. Высота трудовой поверхности над уровнем пола $h_{\text{рп}} = 0,95$ м, окна начинаются с высоты $h_{\text{нок}} = 1$ м, высота окна $h_0 = 2$ м, ширина 4 м. Рабочее пространство находится в IV часовом поясе – в городе Алматы (пояс светового климата – IV северной широты и южнее) Примерное время восхода – 05:49, заката – 19:50. Долгота дня в среднем в летнее время равна 14 часам.

Рабочее пространство находится на расстоянии $l_{\text{рт}} = 1$ м, от наружной стены строения, в которое встроены два оконных проема. Место наименьшей освещенности находится на расстоянии 6 метров от оконного.

Общая площадь окон $S_0, \text{м}^2$, определяется по формулам 5.1 и 5.2

$$100 \frac{S_0}{S_n} = \frac{e_n \cdot K_3 \cdot \eta_0}{\tau_0 r_1}, \quad (5.1)$$

$$S_0 = \frac{S_n \cdot e_n \cdot K_3 \cdot \eta_0}{100 \cdot \tau_0 \cdot r_1}, \quad (5.2)$$

где S_n - площадь помещения; $S_n = 60 \text{ м}^2$;

e_n - нормированное значение КЕО, выбираемое по таблице 1.1. Для высокой точности зрительных работ принимаем $e_n = 1,2$.

m_N - коэффициент светового климата, который находится по таблице 1.1 приложения. Учитывая заданный световой пояс, приняв ориентацию световых проемов на Север, определим по формуле 5.3, при $m_N = 0,75$

$$e_x^{IV} = e_x \cdot m \cdot c, \quad (5.3)$$

где, $m = 0,65$;

$c = 0,75$ (в наружных стенах зданий);

$e_n = 1,2$ для работ высокой точности III разряда зрительной работы;

$$e_x^{IV} = 0,65 \cdot 0,75 \cdot 1,2 = 0,585$$

Учитываем класс здания и находим коэффициент 5,3 $k_3 = 1,2$ (учебные помещения, лаборатории, конструкторские бюро).

τ_0 - общий коэффициент светопропускания, равный

$$\tau_0 = \tau_1 \times \tau_2 \times \tau_3 \times \tau_4, \quad (5.4)$$

где, $\tau_1 = 0,5$ (пустотелые стеклянные блоки);

$\tau_2 = 0,6$ (деревянные двойные раздельные переплеты);

$\tau_3 = 0,8$ (железобетонные фермы и арки);

$\tau_4 = 1$ (регулируемые жалюзи и шторы);

η_0 - световая характеристика окон.

$$\tau_0 = 0,5 \cdot 0,6 \cdot 0,8 \cdot 1 = 0,24$$

Определим по формуле 5.5

$$L = B - 1, \quad (5.5)$$

$$h_{\text{расч}} = h_{\text{ок}} + h_{\text{нок}} - h_{\text{рп}}$$

$$L = 4 - 1 = 3 \text{ м}$$

$$\frac{L}{1} = \frac{L}{B - 1} = \frac{3}{3} = 1$$

$$h_{\text{расч}} = 0,75 + 2 - 0,75 = 2$$

$$\frac{B}{h_{\text{расч}}} = \frac{4}{2} = 2$$

По таблице 5.2 определим $\eta_0 = 9$.

r_1 - коэффициент, учитывающий повышение КЕО при боковом освещении благодаря свету, отраженному от поверхностей помещения и подстилающего слоя.

Средний коэффициент отражения в здании $\rho_{\text{ср}} = 0,5$, принимаем одностороннее боковое освещение. Определяем значение r_1 . Принимаем

r_1 равным 1,05.

$k_{зд}$ – коэффициент, учитывающий затенение окон противостоящими зданиями.

Поскольку затеняющих зданий поблизости нет, $k_{зд} = 1$. Проведем подсчет общей площади окон:

$$S_0 = \frac{60 \cdot 0,585 \cdot 9 \cdot 1 \cdot 1,2}{0,24 \cdot 1,05 \cdot 100} = 15,04 \text{ м}^2$$

Площадь световых проемов равна $S_{сп} = 15,04 \text{ м}^2$.

В проделанных расчетах была посчитана площадь боковых световых проемов помещения, выделенным для отдела закупок, которая необходима для создания нормируемой освещенности на рабочих местах для разряда зрительной работы III. Для здания размерами 10 x 6 x 4 м³ значение для площади одного светового проема получилось равным 15,04 м², где высота окна $h_{ок} = 2$ м, а длина окна $l_{ок} = 4$ м. Можно подытожить, что фактический размер светового проема соответствует нормативным значениям.

5.3 Расчет систем пожаротушения

Таблица 5.3.1 - Группы помещений

Группа помещений	Перечень характерных помещений, производств, технологических процессов
1	Помещения книгохранилищ, библиотек, цирков, хранения сгораемых музейных ценностей, фондохранилищ, музеев и выставок, картинных галерей, концертных и киноконцертных залов, ЭВМ, магазинов, зданий управлений, гостиниц, больниц

Таблица 5.3.2 – Исходные данные для группы помещения 1.

Группа помещений		1
Интенсивность орошения защищаемой площади, л/(с·м ²), не менее	Водой	0,08
	Раствором пенообразователя	-
Расход, л/с, не менее	Воды	10
	Раствора пенообразователя	-
Минимальная площадь спринклерной АУП, м ² , не менее		60
Продолжительность подачи воды, мин, не менее		30
Максимальное расстояние между спринклерными оросителями, м		4

Таблица 5.3.3 – Технические характеристики СВУ

Коэффициент производительности	0,60	0,77	0,84
Диапазон рабочего давления, МПа	0,05 – 1,00		
Защищаемая площадь, м ²	12		
Интенсивность орошения при 0,1(0,3) МПа	0,095(0,175)	0,120(0,200)	0,145(0,215)
Габариты	50x30x27		
Масса, не более, кг	0.055		
Присоединительная резьба	R 1/2		
Термочувствительный элемент	DI 937		
Коэффициент тепловой инерционности	>80		
Номинальная температура срабатывания, °С	68		
Номинальное время срабатывания, с	300		
Предельно допустимая рабочая температура, °С	50		
Цвет жидкости в колбе	Красный		
К-фактор, GPM/PSI (LPM/bar ^{0.5})	8,0 (115)	10,1 (146,1)	11,0 (160)

Расчет расхода воды через оросителя СВУ с коэффициентом производительности 0,77:

$$q_1 = 10K\sqrt{P} \quad (5.10)$$

$$q_1 = 10 \times 0,77\sqrt{0,1} = 2,43 \text{ л/с}$$

где q_1 - расход оросителя, л/с;

K - коэффициент производительности оросителя, принимаемый по технической документации на изделие, л/(с×МПа^{0,5});

P - давление перед оросителем, МПа.

Т.к. площадь помещения составляет 60 м², а один спринклер рассчитан на 12 м², то необходимо установить 5 спринклеров:

$$n = \frac{60}{12} = 5$$

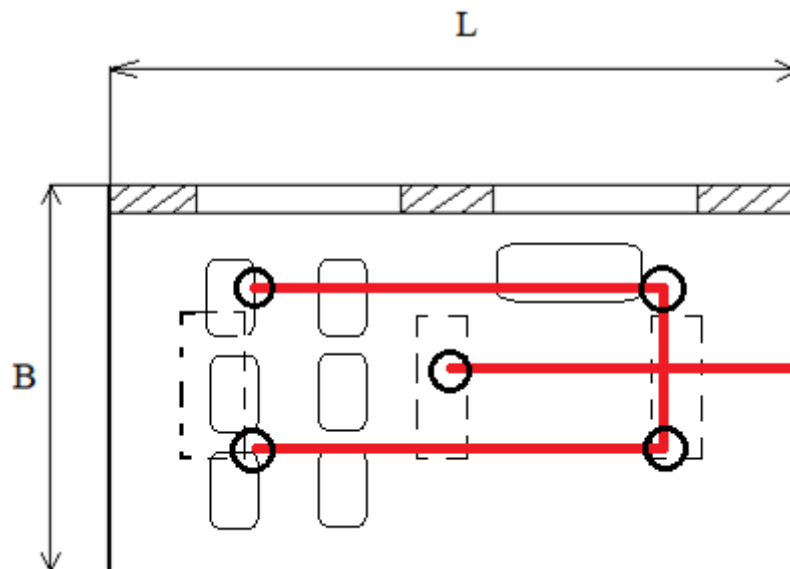


Рисунок 5.2 – Схема размещения оросителей

5.4 Расчет систем кондиционирования рабочего помещения складского персонала

Расчет систем кондиционирования необходим для приведения в соответствие температурного режима помещений нормам и создать благоприятные условия для персонала предприятия.

Выводимый воздух из пространства $t_{\text{ВЫТ}}$, °С,

$$t_{\text{ВЫТ}} = t_{\text{РЗ}} + \Delta t \cdot (h - z) \quad (5.11)$$

где $t_{\text{РЗ}}$ – оптимальная температура для работы ($t_{\text{РЗ}} \leq t_{\text{ДОП}}$), °С;

Для теплого периода года $t_{\text{РЗ}} = 23$ °С;

h - расстояние от пола до вытяжки кондиционера на потолке $h = 4$ м;

Δt - температурный градиент по высоте помещения, °С;

z - высота рабочего помещения, $z = 3,7$ м.

$$t_{\text{ВЫТ}} = 23 + 1,4 \times (4 - 3,7) = 23,4 \text{ °С}$$

Итак, видим, что присутствует избыток явной теплоты. Поэтому для регулировки температурного режима и поддержке на благоприятном уровне, требуется подавать воздух ($t_{\text{ПР}}$) на 5-7°С ниже температуры воздуха в пространстве.

$$t_{\text{ПР}} = 23 - 5 = 18 \text{ °С}$$

Произвели калькуляцию количества приточного воздуха $L_{\text{ПР}}$, м³/ч:

$$L_{\text{ПР}} = \frac{Q_{\text{ИЗБ}}}{c \cdot \rho_{\text{ПР}} \cdot (t_{\text{ВЫТ}} - t_{\text{ПР}})} \quad (5.12)$$

где $\rho_{\text{ПР}}$ - 1,2 кг/м³ - плотность поступающего в помещение воздуха;

c - 1кДж/(кг × °С) – удельная теплоемкость воздуха при постоянном давлении;

$t_{\text{ВЫТ}}$ - температура воздушной массы, удаляемой из рабочего пространства, °С;

$Q_{\text{ИЗБ}}$ – избыточное выделение явной теплоты, кДж/ч;

$t_{\text{ПР}}$ - температура приточного воздуха в рабочее помещение, °С.

Установили величину излишек выделения явной теплоты $Q_{\text{ИЗБ}}$:

$$Q_{\text{ИЗБ}} = \sum Q - \sum Q_{\text{УХ}} \quad (5.13)$$

где $\sum Q$ - суммарное количество поступающей явной теплоты из вне и выделяемой явной теплоты внутри помещения;

$\sum Q_{\text{УХ}}$ - суммарное количество выводящейся из пространства теплоты.

Тепловыделения от искусственного освещения Q_2 :

$$Q_2 = 1000 \times N \quad (5.14)$$

N – Мощность освещения, Вт.

$$Q_2 = 1200 \times 0,25 \times 20 = 6000 \text{ Вт}$$

Тепловыделение от людей Q_3 :

$$Q_3 = n \times q \quad (5.15)$$

где n - число сотрудников;

q - количество тепла, от каждого человека, Вт.

Таблица 5.2 - Количество тепла, от одного человека в зависимости от категории работ и температуры окружающей среды.

Категория выполняемых работ	Выделяемое тепло			
	Полное		Явное	
	10 °С	35 °С	10 °С	35 °С
Легкая	160 Вт	140 Вт	155 Вт	135 Вт

$$Q_3 = 7 \times 140 = 980 \text{ Вт}$$

Количество тепла от солнечной радиации $Q_{\text{ОСТ.РАД}}$:

$$Q_{\text{ОСТ.РАД}} = F_{\text{ОСТ}} \times q_{\text{ОСТ}} \times A_{\text{ОСТ}} \quad (5.16)$$

Количество тепла от солнечной радиации в зависимости от наружного покрытия:

$$Q_{\text{П.РАД}} = F_n \times q_n \times k_n \quad (5.17)$$

где $F_{\text{ОСТ}}$ и F_n - площадь через которое поступает солнечная радиация, м²;
 $q_{\text{ОСТ}}$ и q_n - теплоступления через 1 м² поверхности. коэффициент теплопередачи равен 1 Вт/(м² × °С);

$A_{\text{ОСТ}}$ - коэффициент остекления рабочего помещения;

k_n - коэффициент теплопередачи покрытия, Вт/(м² × °С).

$q_{\text{ОСТ}}$ - принимается в пределах 70 – 210;

$A_{\text{ОСТ}}$ – принимается в пределах 0,25 - 1,25.

$$F_{\text{ОСТ}} = 54 \text{ м}^2$$

$q_{\text{ОСТ}} = 150 \text{ Вт}/(\text{м}^2 \times \text{°С});$

$A_{\text{ОСТ}} = 0,4$ – окна в пространстве не затемненные;

$$Q_{\text{ОСТ.РАД}} = 54 \times 150 \times 0,4 = 3\,240 \text{ Вт}$$

$Q_{\text{ПРАД}} = 18 \text{ Вт}$ - среднее тепlopоступление покрытия;

$$Q_{\text{П.РАД}} = F_n \times q_n \times k_n$$
$$Q_{\text{П.РАД}} = 54 \times 1 \times 18 = 972 \text{ Вт}$$

Количество тепла которое выходит из рабочего пространства $Q_{\text{УХ}}$, кВт:

$$Q_{\text{УХ}} = \frac{\lambda \cdot S \cdot (t_{\text{ВЫТ}} - t_{\text{ПР}})}{\delta} \quad (5.18)$$

где λ - теплопроводность стен, Вт/(м × °С);

S - площадь, м²;

δ - толщина стен, м;

$\delta = 0,3\text{м}$ толщина стен, которые выполнены из бетона К750 с теплопроводностью 1,2 Вт/(м × °С).

$$Q_{\text{УХ}} = \frac{1,2 \cdot 330 \cdot (23,4 - 18)}{0,3} = 7128 \text{ Вт}$$

Сумма входящей теплоты в помещение:

$$\sum Q = Q_2 + Q_3 + Q_{\text{ОСТ.РАД}} + Q_{\text{П.РАД}} \quad (5.19)$$
$$\sum Q = 6000 + 980 + 3240 + 97 = 10317 \text{ Вт}$$

Избыточное выделение явной теплоты:

$$Q_{\text{ИЗБ}} = 10317 - 7128 = 3189$$

Количество приточного воздуха необходимого для поддержания комфортных условий:

$$L_{\text{пр}} = \frac{3189}{1 \cdot 1,2 \cdot (23,4 - 18)} = 492,12 \frac{\text{м}^3}{\text{ч}}$$

Так как присутствует избыточного выделения теплоты, произвели калькуляцию, что приведения нынешних условий труда в благоприятные и соответствию нормативам необходимо произвести монтаж кондиционера.

Наш выбор упал на кассетный кондиционер LG UU24W - инверторный кассетный кондиционер с сплит – системой и возможностью как обогрева, так и охлаждения.

Таблица 5.3 Характеристики сплит-системы GREE-18 R410A

Мощность в режиме охлаждения	5200 Вт
Мощность в режиме обогрева	5500 Вт
Потребляемая мощность при охлаждении	1650 Вт
Потребляемая мощность при обогреве	1550 Вт
Уровень шума (мин/макс)	41/42 Дб
Обслуживаемая площадь	65 кв.м
Внутренний блок сплит-системы	700×600×275 мм
Наружный блок сплит-системы	950×700×396 мм

Заключение

В ходе выполнения дипломной проекта был разработан программный продукт, который повышает продуктивность работы служащих и повышает уровень их организованности на складе малого предприятия. Данное программное решение позволяет сократить персонал, благодаря удачной автоматизации процессов, выполнявшихся до внедрения вручную работниками.

В ходе разработки программного комплекса были использованы следующие программные решения:

- для разработки клиентского интерфейса: 1С: Предприятие 8.3;
- для соединения базы данных между собою использовался: СУБД MSSQL.

В дипломный проект входит введение, 5 глав и итоговое заключение.

Во введении раскрывается актуальность выбранной темы, ставится цель разработки и задачи, которые необходимо выполнить, определяется область применения разработки.

В первой главе производится анализ и выбор технологий для решения задачи.

Во второй главе описывается проектирование приложения.

Третья глава представляет собой описание разработки приложения, также имея при себе описание итогового программного продукта, включая все экранные формы и примеры взаимодействия пользователя с программой.

В четвертой главе производится обоснование экономической целесообразности разрабатываемого проекта.

В пятой главе рассматриваются мероприятия по улучшению условий труда в рамках реализуемого проекта.

Заключение освещает результаты исследования и разработки:

- изучены современные средства разработки клиент-серверных веб-приложений и выполнена разработка согласно поставленной цели;
- разработана дальнейшая стратегия улучшения приложения до полноценного продукта.

Список литературы

- 1 1С:Предприятие 8.3. Практическое пособие разработчика /Авторы: Радченко Максим Григорьевич, Хрусталева Елена Юрьевна. , 2017 г.
- 2 Решение специальных прикладных задач в «1С:Предприятии 8.2» /Авторы: Гончаров Д.И., Хрусталева Е.Ю, 2012 г.
- 3 Профессиональная разработка в системе 1С:Предприятие 8.3" / Авторы: Ажеронок В.А., Габец А.П., Гончаров Д.И., Козырев Д.В., Кухлевский Д.С., Островерх А.В., Радченко М.Г., Хрусталева Е.Ю. , 2012 г.
- 4 Разработка управляемого интерфейса /Авторы: Ажеронок В.А., Островерх А. В., Радченко М. Г., Хрусталева Е. Ю. , 2010 г.
- 5 Разработка сложных отчетов в «1С:Предприятии 8.3». Система компоновки данных /Авторы: Хрусталева Е.Ю. , 2012 г.
- 6 Технологии интеграции 1С:Предприятия/ Авторы: Д. И. Гончаров, Е. Ю. Хрусталева, 2011 г.
- 7 1С:Предприятие 8. Конвертация данных: обмен данными между прикладными решениями /Авторы книги: Бояркин В.Э. Филатов А.И. , 2010 г.
- 8 Облачные технологии «1С:Предприятия»/ Хрусталева Е. Ю, 2016 г.
- 9 Разработка сложных отчетов в «1С:Предприятии 8.3». Система компоновки данных /Авторы: Хрусталева Е.Ю. , 2016 г.
- 10 1С:Предприятие 8. Конвертация данных: обмен данными между прикладными решениями /Авторы книги: Бояркин В.Э. Филатов А.И. , 2010 г.
- 11 Решение специальных прикладных задач в «1С:Предприятии 8.2» /Авторы: Гончаров Д.И., Хрусталева Е.Ю, 2012 г.
- 12 Разработка управляемого интерфейса /Авторы: Ажеронок В.А., Островерх А. В., Радченко М. Г., Хрусталева Е. Ю. , 2010 г.
- 13 Решение специальных прикладных задач в «1С:Предприятии 8.2» /Авторы: Гончаров Д.И., Хрусталева Е.Ю, 2012 г.
- 14 Облачные технологии «1С:Предприятия»/ Хрусталева Е. Ю, 2016 г.
- 15 1С:Предприятие 8. Конвертация данных: обмен данными между прикладными решениями /Авторы книги: Бояркин В.Э. Филатов А.И. , 2011 г.
- 16 1С:Предприятие 8.3. Практическое пособие разработчика /Авторы: Радченко Максим Григорьевич, Хрусталева Елена Юрьевна, 2016 г.
- 17 Облачные технологии «1С:Предприятия»/ Хрусталева Е. Ю, 2016 г.
- 18 Решение специальных прикладных задач в «1С:Предприятии 8.2» /Авторы: Гончаров Д.И., Хрусталева Е.Ю, 2012 г.
- 19 1С:Предприятие 8. Конвертация данных: обмен данными между прикладными решениями /Авторы книги: Бояркин В.Э. Филатов А.И., 2012 г.
- 20 Решение специальных прикладных задач в «1С:Предприятии 8.2» /Авторы: Гончаров Д.И., Хрусталева Е.Ю, 2012 г.
- 21 <https://www.pavlodar.com/zakon/?all=all&dok=02990>

- 22 Графкина, М.В. Охрана труда в непроизводственной сфере: учебное пособие / М. В. Графкина. – Москва: Форум, 2013. – 319 с.
- 23 https://online.zakon.kz/document/?doc_id=1038581
- 24 ГОСТ 12.1.004-91 «Система стандартов безопасности труда.»
- 25 Вершина, Г.А. Охрана труда: учебник / Г. А. Вершина и др. – Минск: ИВЦ Минфина, 2017. – 412 с.
- 26 <http://www.grandars.ru/shkola/bezopasnost-zhiznedeyatelnosti/vrednye-factory-pri-rabote-na-pk.html>

Приложение А (Техническое задание)

Программа: 1С: Предприятие

Краткое описание: Создание конфигурации для ускорения работы в 1С

Постановка задачи:

1) Разработка главных подсистем

2) В подсистеме «Продажи» создать:

- банковскую выписку;
- клиентский заказ;
- реализация ТМЗ;
- указание цены;
- распечатка счета для оплаты заказа;
- статус сделки(заморожена, осуществлена, в разработке);

Реализовать ТМЗ с учетом всех сопутствующих документов и отчетов.

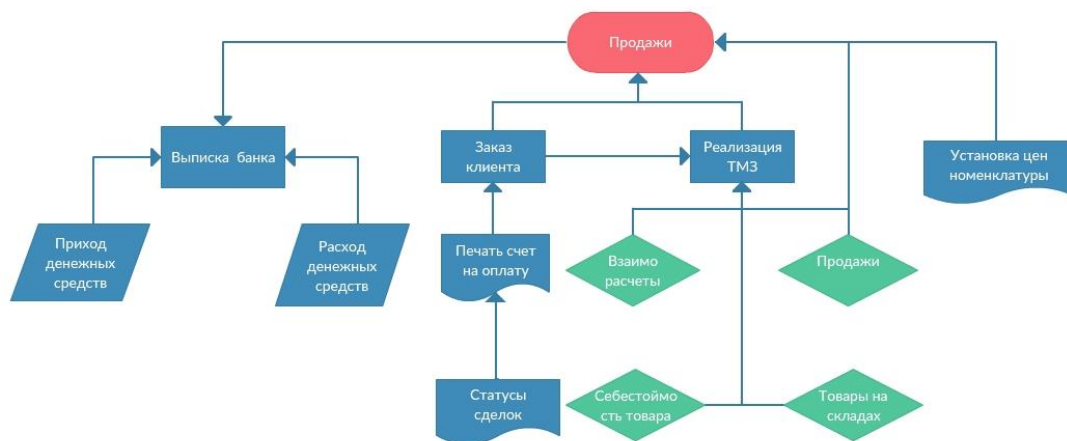


Рисунок 1 – продажи

1) Внести подсистему «Закупки»:

- вхождение ТМЗ;
- документация торговли;

Продолжение приложения А

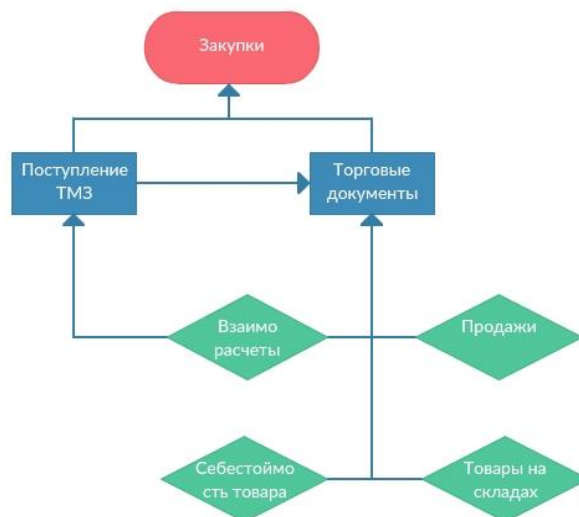


Рисунок 2 – закупки

1) В подсистеме «Зарплата» связать справочники:

- начисление зарплаты;
- невыход сотрудника;
- принятие графика;
- график работ;
- начисление;
- реализация выплаты заработной платы;
- расчет премиальной выплаты;

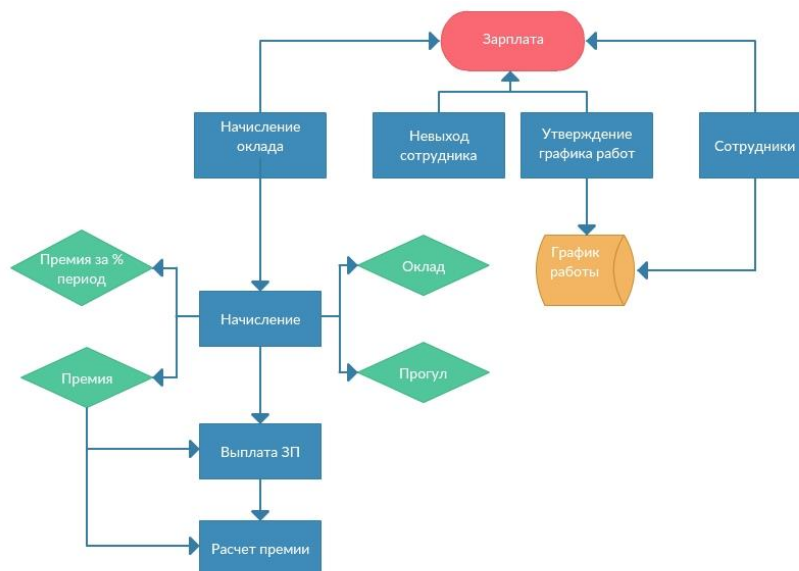


Рисунок 3 - Зарплата

Продолжение приложения А

- 1) Внесение новой подсистемы «Бизнес процессы»:
- задачи настойчивых продаж;
 - задачи приема сотрудника;
 - настойчивая продажа;
 - прием сотрудника на работу;
 - точка маршрута;
 - старт, сбор данных о клиенте, коммерческое предложение, условие продаж, подписание договора, отгрузка товаров, завершение;
 - старт, подготовка рабочего места, знакомство с офисом, знакомство с коллективом, завершение;

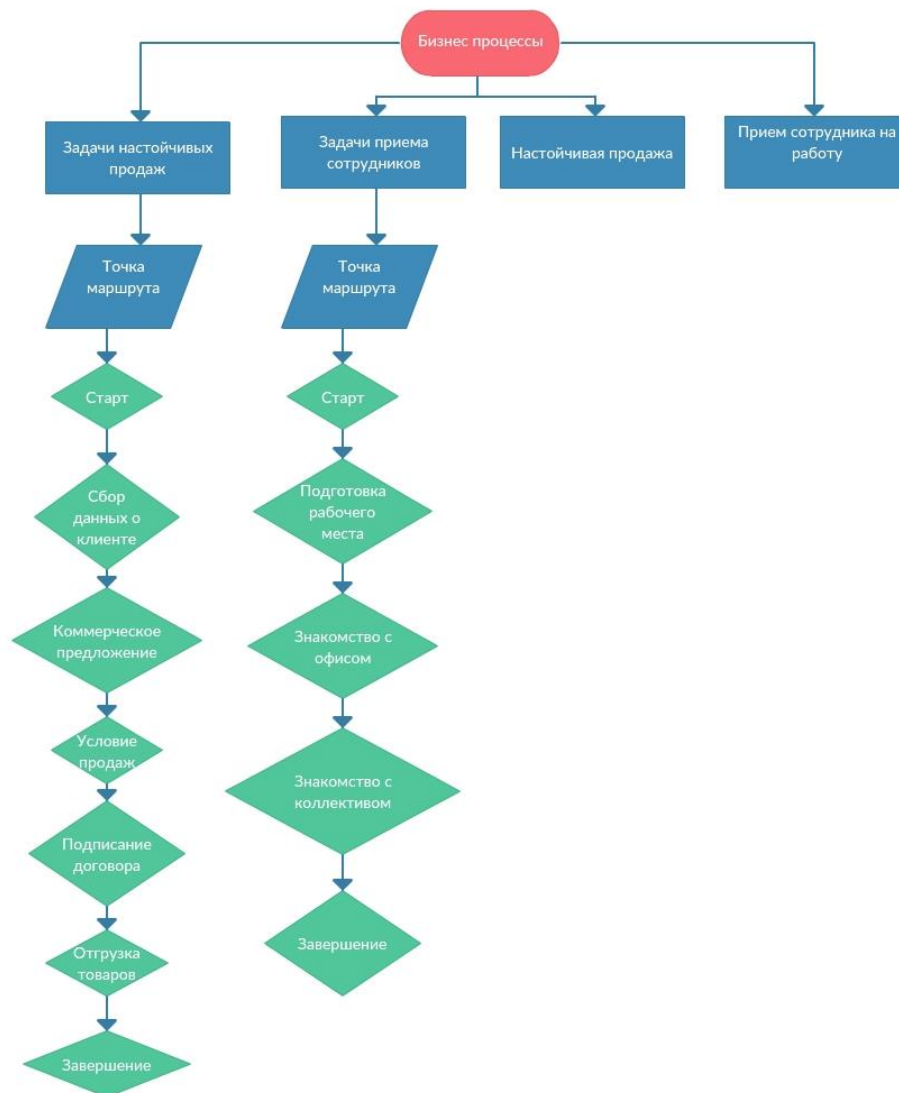


Рисунок 4 – Бизнес процессы

Приложение Б (Листинг программы)

```
#Область ОписаниеПеременных
// СтандартныеПодсистемы
// Хранилище глобальных переменных.
//
// ПараметрыПриложения - Соответствие - хранилище переменных, где:
// * Ключ - Строка - Наименование переменной в формате
"НаименованиеБиблиотеки.НаименованиеПеременной";
// * Значение - Произвольный - значение переменной.
//
// Инициализация (на примере СообщенияДляЖурналаРегистрации):
//                                     НаименованиеПараметра      =
"СтандартныеПодсистемы.СообщенияДляЖурналаРегистрации";
//     Если ПараметрыПриложения[НаименованиеПараметра]      =
Неопределено Тогда
//     ПараметрыПриложения.Вставить(НаименованиеПараметра, Новый
СписокЗначений);
//     ОкончаниеЕсли;
//
// Использование (на примере СообщенияДляЖурналаРегистрации):
//
ПараметрыПриложения["СтандартныеПодсистемы.СообщенияДляЖурналаРе
гистрации"].Добавить(...);
//
ПараметрыПриложения["СтандартныеПодсистемы.СообщенияДляЖурналаРе
гистрации"] = ...;
Перем ПараметрыПриложения Экспорт;
// Окончание СтандартныеПодсистемы
// ЭлектронныеСчетаФактуры
Перем СеансовыеДанныеЭСФ Экспорт;
// Окончание ЭлектронныеСчетаФактуры
// ТехнологияСервиса
Перем
ОповещениеПриПримененииЗапросовНаИспользованиеВнешнихРесурсовВМ
оделиСервиса Экспорт;
// Окончание ТехнологияСервиса
//РаботаСВнешнимОборудованием
Перем глПодключаемоеОборудование Экспорт; // для кэширования на
клиенте
Перем глДоступныеТипыОборудования Экспорт;
//Окончание РаботаСВнешнимОборудованием
```

Продолжение приложения Б

```
#ОкончаниеОбласти
#Область ОбработчикиСлучаев
Процедура ПередНачаломРаботыСистемы()
    // СтандартныеПодсистемы
    СтандартныеПодсистемыКлиент.ПередНачаломРаботыСистемы();
    // Окончание СтандартныеПодсистемы
    // ИнтернетПоддержкаПользователей
ИнтернетПоддержкаПользователейКлиент.ПередНачаломРаботыСистем
ы();
    // Окончание ИнтернетПоддержкаПользователей
    // ПодключаемоеОборудование
    МенеджерОборудованияКлиент.ПередНачаломРаботыСистемы();
    // Окончание ПодключаемоеОборудование
    ОкончаниеПроцедуры
Процедура ПриНачалеРаботыСистемы()
    // СтандартныеПодсистемы
    СтандартныеПодсистемыКлиент.ПриНачалеРаботыСистемы();
    // Окончание СтандартныеПодсистемы
    //РаботаСВнешнимОборудованием
    МенеджерОборудованияКлиент.ПриНачалеРаботыСистемы();
    //Окончание РаботаСВнешнимОборудованием
    ОкончаниеПроцедуры
Процедура                                ПередЗавершениемРаботыСистемы(Отказ,
ТекстПредупреждения)
    // СтандартныеПодсистемы
    СтандартныеПодсистемыКлиент.ПередЗавершениемРаботыСистем
ы(Отказ, ТекстПредупреждения);
    // Окончание СтандартныеПодсистемы
    // ПодключаемоеОборудование
    МенеджерОборудованияКлиент.ПередЗавершениемРаботыСистем
ы();
    // Окончание ПодключаемоеОборудование
    ОкончаниеПроцедуры
Процедура ОбработкаВнешнегоСлучая(Источник, Случай, Данные)
    // ПодключаемоеОборудование
    // Подготовить данные
    ОписаниеСлучая = Новый Структура();
    ОписаниеОшибки = "";
    ОписаниеСлучая.Вставить("Источник", Источник);
    ОписаниеСлучая.Вставить("Случай", Случай);
    ОписаниеСлучая.Вставить("Данные", Данные);
```

Продолжение приложения Б

```
// Передать на обработку данные.
Результат
МенеджерОборудованияКлиент.ОбработатьСлучайОтУстройства(ОписаниеС
лучая, ОписаниеОшибки);
Если Не Результат Тогда
    ОбщегоНазначенияКлиентСервер.СообщитьПользователю(НСтр("ru='П
ри обработке внешнего Случая от устройства произошла ошибка.'")
    + Символы.ПС + ОписаниеОшибки);
    ОкончаниеЕсли;
    // Окончание ПодключаемоеОборудование
ОкончаниеПроцедуры
#ОкончаниеОбласти
#Если Сервер Или ТолстыйКлиентОбычноеПриложение Или
ВнешнееСоединение Тогда
    #Область СлужбныеПроцедурыИФункции
    // Установить или снять блокировку информационной базы,
    // исходя из значений реквизитов обработки.
    Процедура ВыполнитьУстановку() Экспорт
        ВыполнитьУстановкуБлокировки(ЗапретитьРаботуПользователей)
;
    ОкончаниеПроцедуры
    // Отменить ранее установленную блокировку сеансов.
    //
    Процедура ОтменитьБлокировку() Экспорт
        ВыполнитьУстановкуБлокировки(Ложь);
    ОкончаниеПроцедуры
    // Зачитать параметры блокировки информационной базы
    // в реквизиты обработки.
    //
    Процедура ПолучитьПараметрыБлокировки() Экспорт
        Если Пользователи.ЭтоПолноправныйПользователь(, Истина)
Тогда
        ТекущийРежим = ПолучитьБлокировкуСеансов();
        КодДляРазблокировки = ТекущийРежим.КодРазрешения;
    Иначе
        ТекущийРежим
СоединенияИБ.ПолучитьБлокировкуСеансовОбластиДанных());
    ОкончаниеЕсли;
    ЗапретитьРаботуПользователей = ТекущийРежим.Установлена;
```


Продолжение приложения Б

```
СообщениеДляПользователей                                     =
СоединенияИБКлиентСервер.ИзвлечьСообщениеБлокировки(ТекущийРежим.
Сообщение);
    Если ЗапретитьРаботуПользователей Тогда
        НачалоДействияБлокировки = ТекущийРежим.Начало;
        ОкончаниеДействияБлокировки =
ТекущийРежим.Окончание;
        Иначе
            //В случае отсутствия блокировки можем предположить, что
            // пользователь раскрыл форму для внесения блокировки.
            // Исходя из этого ставим дату блокировки соответствующий
текущей дате.
            НачалоДействияБлокировки = НачалоМинуты(ТекущаяДатаСеанса()
+ 5 * 60);
            ОкончаниеЕсли;
            ОкончаниеПроцедуры
Процедура ВыполнитьУстановкуБлокировки(Значение)
    Если Пользователи.ЭтоПолноправныйПользователь(,
Истина) Тогда
        Блокировка = Новый БлокировкаСеансов;
        Блокировка.КодРазрешения = КодДляРазблокировки;
        Иначе
            Блокировка =
СоединенияИБ.НовыеПараметрыБлокировкиСоединений();
            ОкончаниеЕсли;

            Блокировка.Начало = НачалоДействияБлокировки;
            Блокировка.Окончание = ОкончаниеДействияБлокировки;

            Блокировка.Сообщение =
СоединенияИБ.СформироватьСообщениеБлокировки(СообщениеДляПользов
ателей,
                КодДляРазблокировки);
            Блокировка.Установлена = Значение;
            Если Пользователи.ЭтоПолноправныйПользователь(, Истина)
Тогда
                УстановитьБлокировкуСеансов(Блокировка);
            Иначе
                СоединенияИБ.УстановитьБлокировкуСеансовОбластиДанных(Бл
окировка);
            ОкончаниеЕсли;
```

Продолжение приложения Б

```
ОкончаниеПроцедуры
#ОкончаниеОбласти
#ОкончаниеЕсли
#Если Сервер Или ТолстыйКлиентОбычноеПриложение Или
ВнешнееСоединение Тогда
////////////////////////////////////
// СЛУЖЕБНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ
////////////////////////////////////
// Процедуры и функции выгрузки данных отчета в XML
//Выгружает данные отчета в формат XML
// Параметры:
// РевизияВыгрузки - ревизия выгрузки файла, определяет структуру
файла
//
Функция ВыгрузитьОтчетВXMLДляСОНО(ДокументДляВыгрузки,
ЗНАЧ РевизияВыгрузки) Экспорт
    ДанныеДляВыгрузки =
ЗаполнениеДанныхДляВыгрузки(ДокументДляВыгрузки);
    Префикс = РевизияВыгрузки;
    НаименованиеФайла =
ПолучитьНаименованиеВременногоФайла("xml");
    // Запишем общие атрибуты (code, version) и создадим элемент
form.
    Запись = Новый ЗаписьXML;
    Запись.ОткрытьФайл(НаименованиеФайла,"UTF-8");
    Запись.ЗаписатьОбъявлениеXML();
        // Запишем корневой элемент.
    Запись.ЗаписатьНачалоЭлемента("fno");
        МакетВыгрузки = ПолучитьМакет("ВыгрузкаВXML");
    ТаблОбщиеРеквизиты = Новый ТаблицаЗначений;
    ТаблОбщиеРеквизиты.Колонки.Добавить("НаименованиеАтрибут
a");
        ТаблОбщиеРеквизиты.Колонки.Добавить("ЗначениеАтрибута");
    ОбластьОбщихРеквизитов =
МакетВыгрузки.Область("ОбщиеРеквизиты" + Префикс);
        Для Ном = ОбластьОбщихРеквизитов.Верх По
ОбластьОбщихРеквизитов.Низ Цикл
            Строка = ТаблОбщиеРеквизиты.Добавить();
            Строка.НаименованиеАтрибута =
СокрП(МакетВыгрузки.Область(Ном, 1).Текст);
```

Продолжение приложения Б

```
Строка.ЗначениеАтрибута = СокрП(МакетВыгрузки.Область(Ном,  
2).Текст);  
    ОкончаниеЦикла;  
        // Запишем атрибуты корневого элемента.  
Для Каждого Строка Из ТаблОбщиеРеквизиты Цикл  
    Запись.ЗаписатьАтрибут(Строка.НаименованиеАтрибута,  
Строка.ЗначениеАтрибута);  
    ОкончаниеЦикла;  
    // Создадим таблицу для выгрузки.  
ТаблицаВыгрузки = Новый ТаблицаЗначений;  
ТаблицаВыгрузки.Колонки.Добавить("Форма");  
ТаблицаВыгрузки.Колонки.Добавить("Страница");  
ТаблицаВыгрузки.Колонки.Добавить("НаименованиеАтрибута");  
ТаблицаВыгрузки.Колонки.Добавить("НаименованиеЯчейки");  
ТаблицаВыгрузки.Колонки.Добавить("КоличествоROW");  
ТаблицаВыгрузки.Колонки.Добавить("ГруппаROW");  
ТаблицаВыгрузки.Колонки.Добавить("НаименованиеФормы");  
ТаблицаВыгрузки.Колонки.Добавить("НаименованиеЛиста");  
ТаблицаВыгрузки.Колонки.Добавить("Отображение");  
ТаблицаВыгрузки.Колонки.Добавить("КодФормы");  
ТаблицаВыгрузки.Колонки.Добавить("Выгружать");  
ТаблицаВыгрузки.Колонки.Добавить("Многострочность");  
ТаблицаВыгрузки.Колонки.Добавить("ЗаголовокФормы");  
ТаблицаВыгрузки.Колонки.Добавить("КоличествоНаЛисте");  
ТаблицаВыгрузки.Колонки.Добавить("Значение");  
ОбластьВыгрузки = МакетВыгрузки.Область("Выгрузка" +  
Префикс);  
ТекущееНаименованиеФормы = Неопределено;  
Для Ном = ОбластьВыгрузки.Верх По ОбластьВыгрузки.Низ Цикл  
    Строка = ТаблицаВыгрузки.Добавить();  
    Строка.Форма = СокрП(МакетВыгрузки.Область(Ном, 1).Текст);  
        Строка.Страница =  
СокрП(МакетВыгрузки.Область(Ном, 2).Текст);  
        Строка.НаименованиеАтрибута =  
СокрП(МакетВыгрузки.Область(Ном, 3).Текст);  
        Строка.НаименованиеЯчейки =  
СокрП(МакетВыгрузки.Область(Ном, 4).Текст);  
        Строка.КоличествоROW =  
СокрП(МакетВыгрузки.Область(Ном, 5).Текст);  
        Строка.НаименованиеФормы =  
СокрП(МакетВыгрузки.Область(Ном, 6).Текст);
```

Продолжение приложения Б

Строка.НаименованиеЛиста = СокрП(МакетВыгрузки.Область(Ном,
7).Текст);

Строка.Отображение =
СокрП(МакетВыгрузки.Область(Ном, 8).Текст);

Строка.ГруппаROW =
СокрП(МакетВыгрузки.Область(Ном, 9).Текст);

НовоеНаименованиеФормы = Строка.НаименованиеФормы;

Если НовоеНаименованиеФормы <> ТекущееНаименованиеФормы

ТогдаПопытка

Макет = ПолучитьМакет("Показатели");

Форма = Макет.ПолучитьОбласть(Строка.НаименованиеФормы +"_" +
Префикс);

Исключение

Форма = Неопределено;

ОкончаниеПопытки;

ОкончаниеЕсли;

Если Форма <> Неопределено Тогда

Строка.КодФормы = СокрП(Форма.Область(1, 1).Текст);

Строка.Выгружать = Число(Форма.Область(1, 4).Текст);

Строка.Многострочность = Число(Форма.Область(1, 6).Текст);

Строка.ЗаголовокФормы = СокрП(Форма.Область(1, 9).Текст);

Строка.КоличествоНаЛисте = Число(Форма.Область(1, 10).Текст);

Строка.Значение = "";

ОкончаниеЕсли;

ТекущееНаименованиеФормы = Строка.НаименованиеФормы;

ОкончаниеЦикла;

// Выгрузка в XML.

КоличествоСтрокВТаблице = ТаблицаВыгрузки.Количество();

НомерСтроки = 0;

ТекущаяФорма = Неопределено;

ТекущаяСтраница = Неопределено;

Пока Истина Цикл

Если НомерСтроки >= КоличествоСтрокВТаблице Тогда

Прервать;

ОкончаниеЕсли;

ВыгрузитьСтрокуВXML(Запись, ТаблицаВыгрузки,
КоличествоСтрокВТаблице, НомерСтроки, ДанныеДляВыгрузки,
ТекущаяФорма, ТекущаяСтраница);

НомерСтроки = НомерСтроки + 1;

ОкончаниеЦикла;

Запись.ЗаписатьОкончаниеЭлемента(); // </sheet>

Продолжение приложения Б

```
Запись.ЗаписатьОкончаниеЭлемента(); // </sheetGroup>
    Запись.ЗаписатьОкончаниеЭлемента(); // </form>
    Запись.ЗаписатьОкончаниеЭлемента(); // fno
    Запись.Закрыть();
    // Данный текст модуля добавлен для совместимости с СОНО (в
СОНО для UTF-8 нет ВОМ поля).
    ТекстДок = Новый ТекстовыйДокумент;
    ТекстДок.Прочитать(НаименованиеФайла, "windows-1251");
    Строка = ТекстДок.ПолучитьСтроку(1);
    ТекстДок.ЗаменитьСтроку(1,Прав(Строка,СтрДлина(Строка)-3));
    ТекстДок.Записать(НаименованиеФайла, "windows-1251");
    ДвоичныеДанныеФайла=НовыйДвоичныеДанные(Наименование
Файла);
    ФайлВременный = Новый Файл(НаименованиеФайла);
    Попытка
        ФайлВременный.УстановитьТолькоЧтение(Ложь);
        УдалитьФайлы(НаименованиеФайла);
    Исключение
        ЗаписьЖурналаРегистрации(НСтр("ru =
'ВыгрузкаVXMLЗаявленияОВвозеТоваров.Удаление файлов при выгрузке",
        ОбщегоНазначенияКлиентСервер.КодОсновногоЯзыка()),
        УровеньЖурналаРегистрации.Ошибка,
        ИнформацияОбОшибке());
    ОкончаниеПопытки;
    Возврат
ПоместитьВоВременноеХранилище(ДвоичныеДанныеФайла);
    ОкончаниеФункции

    // Формирует построчно структуру XML файла из макета
"ВыгрузкаVXML"
    //
    Процедура ВыгрузитьСтрокуVXML(Запись, ТаблицаВыгрузки,
КоличествоСтрокВТаблице, НомерСтроки, ДанныеДляВыгрузки,
ТекущаяФорма = Неопределено, ТекущаяСтраница = Неопределено)
        СтрокаВыгрузки = ТаблицаВыгрузки.Получить(НомерСтроки);
        НоваяФорма = СтрокаВыгрузки.Форма;
        НоваяСтраница = СтрокаВыгрузки.Страница;
        Если ТекущаяФорма <> Неопределено И НоваяФорма <>
ТекущаяФорма Тогда
            Запись.ЗаписатьОкончаниеЭлемента(); // </sheet>
            ОкончаниеЕсли;
```

Продолжение приложения Б

```
Если НЕ НоваяФорма = ТекущаяФорма Тогда
    Если НЕ ТекущаяФорма = Неопределено Тогда
        Запись.ЗаписатьОкончаниеЭлемента(); //
</sheetGroup>
        Запись.ЗаписатьОкончаниеЭлемента(); //
</form>
    ОкончаниеЕсли;
    Запись.ЗаписатьНачалоЭлемента("form");
    Запись.ЗаписатьАтрибут("name",СтрокаВыгрузки.Форма);
    Запись.ЗаписатьНачалоЭлемента("sheetGroup");
    ОкончаниеЕсли;
    Если НЕ НоваяСтраница = ТекущаяСтраница Тогда
        Если ТекущаяФорма <> Неопределено И НоваяФорма <>
ТекущаяФорма Тогда
            // ничего не делаем
        ИначеЕсли НЕ ТекущаяСтраница = Неопределено Тогда
            Запись.ЗаписатьОкончаниеЭлемента(); //
</sheet>
    ОкончаниеЕсли;
    Запись.ЗаписатьНачалоЭлемента("sheet");
    Запись.ЗаписатьАтрибут("name",СтрокаВыгрузки.Страница);
    ОкончаниеЕсли;
    Если СтрокаВыгрузки.Многострочность <> Неопределено И
СтрокаВыгрузки.Многострочность = 1 Тогда
        ВыгрузитьМногострочнуюФорму(Запись,
ТаблицаВыгрузки, КоличествоСтрокВТаблице, НомерСтроки,
ДанныеДляВыгрузки);
        // проверим многострочное ли значение
        ИначеЕсли
ЗначениеЗаполнено(СтрокаВыгрузки.КоличествоROW) И
СтрокаВыгрузки.КоличествоROW <> "0" Тогда // для обычных форм
        // если в обычной форме встретилась многострочность
        КоличествоПовторений =
Число(СтрокаВыгрузки.КоличествоROW);
        ГруппаRow =
СтрокаВыгрузки.ГруппаROW;
        Для Н = 1 По КоличествоПовторений Цикл
            Запись.ЗаписатьНачалоЭлемента("row");
            СтрокаВыгрузки =
ТаблицаВыгрузки.Получить(НомерСтроки);
```

Если $N = 1$ Тогда
ТаблицаМногострочныхТегов
ТаблицаВыгрузки.СкопироватьКолонки();

=

Приложение В Акт внедрения

ТОО «RESTOFOOD(РЕСТОФУД)»
БИН 200240017776
Республика Казахстан,
г. Алматы, ул.Алатау 45
Тел. +7(705)220-42-03
e-mail: info@restofood.kz
АО "Евразийский банк"
KZ2994806KZT22035829
БИК EURIKZKA

Resto Food

«RESTOFOOD(РЕСТОФУД)» ЖШС
БСН 200240017776
Қазақстан Республикасы, Алматы қ.,
Алатау үй 45.
Тел. +7(705)220-42-03
e-mail: info@restofood.kz
«Еуразиялық банк» АҚ
KZ2994806KZT22035829
БИК EURIKZKA

Утверждаю
Директор ТОО
«RESTOFOOD(РЕСТОФУД)»
Нурлан Е.Н.
«5» мая 2020г.

АКТ ВНЕДРЕНИЯ

Настоящий акт составлен о том, что результаты выпускной работы студента НАО «АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА» группы ВТ-16-2 очной формы обучения Нурланулы М. на тему «Разработка системы учета движения товаров на складе малого предприятия» внедрены в ТОО «RESTOFOOD(РЕСТОФУД)» и используются в компании для формирования отчетов о движении товаров во всех имеющихся складах. Использование результата выпускной работы Нурланулы М. обеспечивает доступ к детализированным отчетам о движении товаров, с возможностью детализации по отдельным типам товаров, организациям поставки, складам, актуальности товаров в требуемый отрезок даты.

Директор
ТОО «RESTOFOOD(РЕСТОФУД)»



Нурлан Е.Н.

Нурлан Е.Н.