

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ИНФОРМАЦИОННЫХ И РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

**РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ ПОДСИСТЕМЫ МАГАЗИНА
КОМПЬЮТЕРНОЙ ТЕХНИКИ**

Выпускная квалификационная работа
обучающегося по направлению подготовки 09.03.02 Информационные системы
и технологии
очной формы обучения, группы 07001409
Бондарева Сергея Владимировича

Научный руководитель
к.с.н., доцент
Игрунова С.В.

БЕЛГОРОД 2018

РЕФЕРАТ

Информационная подсистема магазина компьютерной техники – Бондарева С.В., выпускная квалификационная работа бакалавра, Белгород, Белгородский государственный национальный исследовательский университет (НИУ «БелГУ»), количество страниц 73, включая 4 приложения, количество рисунков 45, количество таблиц 5, количество использованных источников 41.

КЛЮЧЕВЫЕ СЛОВА: информационная система, база данных, учет продаж компьютерной техники, складской учет, аналитика, SQL, C++, Firebird, IVExpert, функциональное моделирование, пользовательский интерфейс.

ОБЪЕКТ ИССЛЕДОВАНИЯ: торговая деятельность компьютерного сервиса «RossComService»

ПРЕДМЕТ ИССЛЕДОВАНИЯ: ведение взаимоотношений с клиентами и поставщиками, а также складской учет.

ЦЕЛЬ РАБОТЫ: увеличение величины прибыли компьютерного сервиса «RossComService» за счет разработки и внедрения автоматизированной подсистемы компьютерного магазина.

ЗАДАЧИ ИССЛЕДОВАНИЯ: дать характеристику предприятия и исследовать существующие бизнес-процессы; спроектировать структурно-функциональные модели деятельности «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ»; провести анализ существующих разработок; осуществить обоснование проектных решений по техническому, информационному, программному и технологическому обеспечению задачи; разработать и протестировать программное средство; оценить экономическую эффективность разработки.

МЕТОДЫ ИССЛЕДОВАНИЯ: анализ литературы; анализ нормативно-правовой документации торговой деятельности в России; функциональное моделирование

ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ: В результате работы была спроектирована и реализована автоматизированная подсистема магазина компьютерной техники с учетом особенностей торговой деятельности.

Для организации эффективной работы пользователей с подсистемой магазина компьютерной техники необходимы три составляющих: программа, установленная на компьютере пользователя, которая может осуществлять сетевой запрос с целью получения объекта, и предназначенная

для его обработки (например, просмотра, изменения или печати документа); программа, установленная, как правило, на компьютере, где расположен информационный объект, которая может осуществлять по запросу поиск и пересылку объекта, а также упорядочивание доступа к нему нескольких пользователей; правила (протокол) взаимодействия между этими программами.

Использовать систему будут пользователи средней и низкой квалификации. Интерфейс системы должен быть максимально приближен к интерфейсам подобных систем. Ввод информации должен осуществляться в наиболее унифицированных формах.

Система должна работать под управлением ОС семейства Win32. СУБД MS SQL Server или Firebird.

Подсистема компьютерного магазина является экономически эффективной, так как чистая прибыль от использования разработанного программного средства за год превышает затраты на использование системы и за 9 месяца покрывает её себестоимость.

В будущем планируется интеграция разработанной подсистемы компьютерного магазина с уже функционирующей системой учета сервисных услуг. Интеграция программных систем и продуктов – это обмен данными между системами с возможной последующей их обработкой.

Кроме того, планируется разработка дополнительных аналитических отчетных форм, которые позволят управленческому звену компании более точно прогнозировать развитие торговой деятельности.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Обследование предметной области	8
1.1 Характеристика предприятия	8
1.2 Экономическая сущность задачи	10
1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи	11
1.4 Цель и назначение разработки.....	16
1.5 Общая характеристика организации решения задачи на ЭВМ.....	17
1.6 Анализ существующих разработок и обоснование выбора технологии проектирования	21
2 Обоснование проектных решений автоматизации.....	25
2.1 Обоснование проектных решений по техническому обеспечению.....	25
2.2 Обоснование проектных решений по информационному обеспечению	26
2.3 Обоснование проектных решений по программному обеспечению	28
2.4 Обоснование выбора программных средств	30
3 Проектная часть	33
3.1 Информационная модель и ее описание.....	33
3.2 Разработка базы данных.....	35
3.3 Разработка пользовательского интерфейса автоматизированной подсистемы.....	40
3.4 Описание контрольного примера реализации проекта	45
3.5 Оценка экономической эффективности разработки	61
ЗАКЛЮЧЕНИЕ.....	68
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	70
ПРИЛОЖЕНИЕ А	75
ПРИЛОЖЕНИЕ Б	77
ПРИЛОЖЕНИЕ В.....	79
ПРИЛОЖЕНИЕ Г	82

ВВЕДЕНИЕ

В настоящее время информационные технологии успешно проникают во все области человеческой деятельности. Они помогают оптимизировать и упростить жизнь людей, автоматизировать и ускорить решение множества задач, различного рода [1].

Одной из важнейших и актуальных задач, стоящих перед руководителем любой организации, является создание автоматизированных рабочих мест сотрудников. Это позволит увеличить эффективность функционирования предприятия, производительность работников, облегчить процессы взаимодействия клиентов и сотрудников, а тем самым увеличить чистую прибыль организации.

Особенно это становится актуальным в сфере торговли, где эффективность работы предприятия напрямую зависит от уровня автоматизации его бизнес-процессов.

Таким образом, объектом исследования является торговая деятельность компьютерного сервиса «RossComService», а предметом - ведение взаимоотношений с клиентами и поставщиками, а также складской учет.

В условия рынка успешность работы предприятия оценивается в величине прибыли, получаемой при реализации товаров.

Целью выпускной квалификационной работы является увеличение величины прибыли компьютерного сервиса «RossComService» за счет разработки и внедрения автоматизированной подсистемы компьютерного магазина.

Величина прибыли зависит от объема, ассортимента реализуемой продукции, уровня издержек производства и уровня цен, по которым она продается. Следовательно, прибыль в условиях рынка является результатом не только успешного использования факторов производства, но и синтетическим показателем всех видов хозяйственно-производственной деятельности.

Одним из способов увеличения объемов прибыли является автоматизация бизнес-процессов взаимодействия с клиентами и поставщика, а также складского учета.

Для достижения поставленной цели необходимо решить ряд задач:

- дать характеристику предприятия и исследовать существующие бизнес-процессы;
- спроектировать структурно-функциональные модели деятельности «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ»;
- провести анализ существующих разработок;
- осуществить обоснование проектных решений по техническому, информационному, программному и технологическому обеспечению задачи;
- разработать и протестировать программное средство;
- оценить экономическую эффективность разработки.

Основными методами исследования являются:

- анализ литературы;
- анализ нормативно-правовой документации торговой деятельности в России;
- функциональное моделирование.

Таким образом, выпускная квалификационная работа состоит из введения, основной части, которая в свою очередь состоит из 3 взаимосвязанных подчастей, заключения, списка литературы и приложения.

Во введении указываются предмет и объект исследования, цели и задачи проекта, а так же актуальность выбранной темы.

Первый раздел содержит технико-экономическую характеристику компьютерного сервиса «RossComService». Кроме того, в нем описаны результаты проведения существующих разработок подобных систем, а также торговая деятельности компьютерного сервиса «RossComService», построена функциональная модель «КАК ЕСТЬ».

Второй раздел посвящён обоснованию проектных решений по техническому, информационному, программному и технологическому обеспечению автоматизации «RossComService».

В третьем разделе описываются этапы проектирования и реализации автоматизированной системы компьютерного магазина, включает в себя функциональную модель «КАК ЕСТЬ», модели базы данных и описание контрольного примера. Кроме того, в разделе проводится оценка экономической эффективности проекта.

Выпускная квалификационная работа написана на 73 листах, содержит 45 рисунков и 5 таблицы.

1 Обследование предметной области

1.1 Характеристика предприятия

«RossComService» – это зарекомендовавшая себя в течение 9 лет компания по ремонту компьютеров на территории города Россоши и Россошанского района.

Сотрудники «RossComService» производят диагностику, ремонт компьютеров и ноутбуков любой сложности и имеют огромный опыт, накопленный за годы работы.

Основным направлением деятельности компании является ремонт и обслуживание компьютерной техники и решение тех проблем, которые могут возникнуть при ее использовании.

Сервисный центр по ремонту ноутбуков обеспечивает высокое качество ремонта ноутбуков, мониторов, компьютеров и гарантирует индивидуальный подход каждому клиенту. Среди других компаний, которые занимаются ремонтом компьютеров и ноутбуков, «RossComService» выгодно отличается качеством работы, сроками выполнения, и стоимостью услуг.

Сервисный центр «RossComService» оказывает полный комплекс услуг по ремонту компьютеров, ноутбуков, мониторов, при котором инженеры проводят полную диагностику неисправностей и их устранение. При ремонте компьютерного оборудования используются оригинальные запчасти от надежных поставщиков.

Сервисный центр «RossComService» – это:

– самое современное оборудование для диагностики и ремонта компьютерного оборудования: от комплексов для восстановления данных до паяльных станций, способных перепаяивать BGA чипы;

- более 20 специалистов, регулярно проходящих обучение и способных произвести приемку, диагностику и ремонт оборудования любой сложности;

- два офиса приема заказов, рабочие помещения (более 200 кв.м.), склад запчастей, а также система управления ремонтами.

- более 1500 выполненных заказов в месяц.

Компания стремится стать для своих клиентов хорошим партнером, который возьмет на себя все проблемы, возникающие при эксплуатации компьютерной техники.

Абонентское обслуживание компьютерных офисов фирм – одно из основных направлений деятельности компьютерного сервиса.

Компания проводит сервисное обслуживание офисной техники, локальных сетей организаций; оказывает услуги по ремонту, настройке, обслуживанию компьютеров в рамках договора на абонентское обслуживание. Задача компьютерного сервиса – предлагать клиентам лучшие комплексные решения по ремонту и настройке компьютерной и офисной техники.

Перед началом работы специалисты проводят диагностику и предлагают различные варианты решения проблемы, от переустановки программного обеспечения до замены неисправных аппаратных комплектующих.

В 2017 году компанией было принято решение расширить основное направление деятельности и заняться также продажей компьютерной техники. Таким образом, новым направлением деятельности компании становится удовлетворение потребностей покупателя в приобретении товаров сферы компьютерной техники и электроники.

Продукция данной сферы не производится на отечественном рынке, следовательно, немаловажную роль снабжения компьютерного магазина играют дистрибьюторы зарубежных фирм-производителей (поставщики).

В свою очередь магазин компьютерной техники осуществляет реализацию продукции, то есть занимается доведением товара до потребителя. Деятельность компании направлена на продажу товаров компьютерной сферы различных производителей, с которыми был заключён договор на право представлять данную марку той или иной компьютерной продукции.

Для компьютерного магазина основными бизнес-процессами являются:

- закупка и поставка продукции в магазин;
- размещение продукции на складах;
- представление имеющейся продукции покупателю;
- продажа компьютерной продукции;
- учет движения товаров на складах.

1.2 Экономическая сущность задачи

Успешность ведения предпринимательской деятельности характеризуется размером прибыли [2].

Прибыль является очень важной категорией в рыночной экономике. В настоящее время условием существования фирм является их рентабельность (получение прибыли) [3].

Эффективность функционирования предприятия, независимо от организационно-правовой формы и видов его деятельности в условиях рынка определяется способностью предприятия приносить достаточный доход или прибыль.

Прибыль – это конечный результат работы предприятия, стимулирующий дальнейшую производственную деятельность и создающий основу для её расширения. Чем больше предприятие реализует рентабельной продукции, тем больше получит прибыли, тем лучше его финансовое состояние [4].

Получение прибыли обычно является главной целью и движущим мотивом всех видов предпринимательства [5].

Экономической сущностью задачи автоматизации деятельности «RossComService» является повышение результативности труда и прибыли посредством автоматизации рутинной деятельности.

Вследствие ускорения процесса делопроизводства возможно будет увеличить количество производимых продаж и тем самым повысить обороты компании, а в конечном итоге и прибыль.

1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи

При автоматизации деятельности компании по продаже компьютерной техники необходимо решить ряд задач, связанных с хранением большего объёма данных, а также оперативного доступа к ним.

Использование автоматизированной системы в такой быстро меняющейся отрасли как торговля поможет быстрее оценивать конъюнктуру рынка и быстро реагировать на произошедшие изменения, что в совокупности даст максимальное количество прибыли [6]. Это очень важно для выживания фирмы на рынке во время кризиса. Позволит оценить издержки.

В будущем же система позволит также оценивать и долгосрочные перспективы и давать информацию для понимания того во что стоит вкладывать деньги на длительную перспективу.

Среди задач автоматизации отдельно следует выделить задачу автоматизации ведения базы номенклатуры компьютерной техники для продажи, а также ведение базы данных клиентов и поставщиков организации [7].

Накопление данных о клиентах, является очень важной частью автоматизации предприятия, которое направлено на торговлю, по этой базе

можно понять, что нужно клиентам в данный момент, даже если мы не можем в данный момент удовлетворить запрос клиента, информация о том, что ему нужно поможет в развитии ассортимента нашей компании. База клиентов легко позволит провести исследование с целью лучшего своих предложений на рынке. Даст преимущество перед конкурентами, у которых отсутствует автоматизация.

Автоматизация торгового предприятия даст возможность быстрее и полнее реагировать на пожелания клиентов, производить сложные выборки и поиски по многочисленным параметрам, что существенно превосходит сложившуюся в данный момент времени ситуацию, когда все данных хранятся в текстовых и табличных документах пакета Microsoft Office.

Проектирование информационной системы является важной частью разработки, так как в этом процессе производится теоретическое исследование различных областей, которые затрагивает создаваемая система.

При проектировании производится исследование области применения и различных способов и методик создания информационных систем, проектирование частей системы [8].

Структурно-функциональный анализ деятельности магазина компьютерной техники позволит понять, каким образом необходимо организовать внутреннее взаимодействие модулей системы. Для этого были разработана функциональная модель деятельности «КАК ЕСТЬ» с использованием методологии методологии SADT (IDEF0) с помощью CASE-средства AllFusion Process Modeler 7 [9].

Методология IDEF0 используется для создания функциональной модели, которая является структурированным изображением функций производственной системы или среды, а также информации и объектов, связывающих эти функции [10].

Под системой понимается любая комбинация средств аппаратного и программного обеспечения, а также людей. При создании новых систем IDEF0 может вначале применяться для определения требований и функций,

а затем для разработки системы, которая удовлетворяет этим требованиям и реализует эти функции. [11]

На диаграммах все функции производственной системы и интерфейсы представлены как блоки и дуги. Управляющие производством данные входят в блок сверху, в то время как информационные потоки, которые подвергаются изменениям, показаны с левой стороны блока, а результаты выхода показаны с правой стороны. Механизм (человек или автоматизированная система), который осуществляет операцию, представляется дугой, входящей в блок снизу.

На рисунке 1.1 представлена контекстная диаграмма деятельности магазина компьютерной техники.

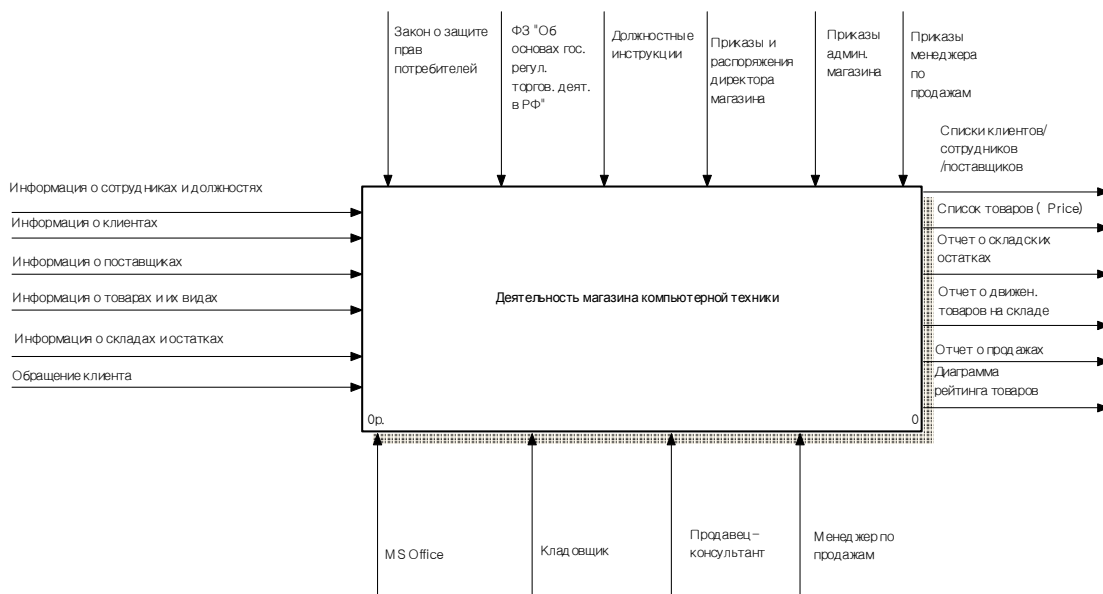


Рисунок 1.1 – Контекстная диаграмма деятельности магазина «КАК ЕСТЬ»

Входными данными являются информация о сотрудниках, клиентах, поставщиках, товарах и их видах, информация о складах и остатках товаров на них, и любой вид обращения клиентов. Управление осуществляется на основе закона о защите прав потребителей, федерального закона «Об основах государственного регулирования торговой деятельности в РФ», должностной инструкции сотрудников магазина, а также приказов директора и

администратора магазина. Механизмами выступают сотрудники магазина: кладовщик и продавец-консультант, а также программное обеспечение в виде пакета MSOffice, в котором в данный момент ведется весь учет продаж компании. Выходные потоки данных представлены в виде списков клиентов, сотрудников, поставщиков, товаров, отчетах о складских остатках и движении товаров на складе, а также о продажах в разрезе товаров с возможностью графического представления показателей.

На рисунке 1.2 представлена диаграмма-декомпозиция деятельности магазина компьютерной техники. В соответствии с бизнес-процессами организации основными функциональными блоками являются: закупка товаров, складской учет, продажа компьютерной техники и анализ торговой деятельности магазина.

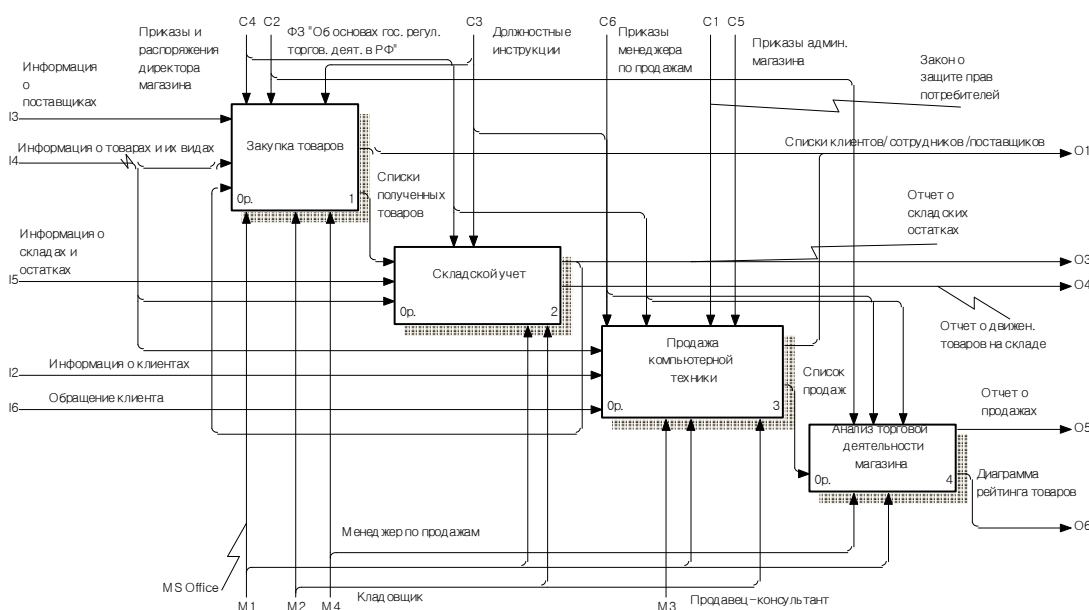


Рисунок 1.2 – Диаграмма-декомпозиция деятельности магазина «КАК ЕСТЬ»

Процесс закупки товара включает в себя анализ складских остатков, формирование плана закупки, подбор поставщика, закупку товара и формирование отчетности. Основными механизмами выступают менеджер по продажам, MS Office, а также кладовщик (рисунок А.1).

Складкой учет представлен в виде 5 подпроцессов: анализ складских резервов, оприходование товара на склад, перемещение товаров между складами, списание товара, формирование отчетности по складам (рисунок А.2).

Оприходование – это операция инвентаризации, при которой значение расчетного остатка приводится к фактическому наличию путем добавления неучтенных товаров. Перемещение товаров на предприятии осуществляется между существующими в компании складами. Списание товаров - это их выбытие, снятие с балансового учета по различным причинам. Все эти операции кладовщик ведет в реестре, который необходим для построения аналитической отчетности. Отчет показывает движения и остатки товаров на складах. Он помогает понять, где и в каком количестве находится товар [12].

Далее идет процесс продажи компьютерной техники, который состоит из следующих подпроцессов: регистрация клиента, подбор товара, оформление заказа, выдача товара, формирование отчетности по продажам. Регистрацией клиентов, подбором и оформлением заказа занимается продавец-консультант, который представлен одним из механизмов. Выдачей товара на основе документа об оплате занимается кладовщик, а формированием отчетности также занимается продавец-консультант. Контроль осуществляется на основе приказов и распоряжений директора магазина, администратора магазина, менеджера по продажам, а также закона о защите прав потребителей (рисунок А.3).

Последним бизнес-процессом является анализ торговой деятельности компании, который осуществляет менеджер по продажам компании на основе всей сформированной и переданной отчетности. Выходными параметрами являются отчет о продажах, а также диаграмма рейтинга услуг (рисунок А.4).

Все описанные выше процессы выполняются с использованием офисного пакета приложений MS Office.

Каждый бизнес-процесс заканчивается формированием отчетности, что является очень длительным и трудозатратным действием для всех участвующих сотрудников из-за большого объема меняющихся данных. Кроме того, возникает множество ошибок в процессе её формирования и отчетность очень быстро становится неактуальной из-за постоянного увеличения объема данных.

1.4 Цель и назначение разработки

Под автоматизацией деятельности магазина компьютерной техники подразумевают решение следующих задач:

- автоматизация учёта постоянных и временных сотрудников организации и их должностях;
- автоматизация учёта поставщиков;
- автоматизация учёта клиентов компании;
- автоматизация учёта товаров (компьютерной техники) и их видов;
- автоматизация процессов движения товаров на складах и анализа загруженности;
- автоматизация учета заказов клиентов;
- автоматизация задачи хранения всей документации и отчётности.

Система должна обеспечивать следующие функции ввод, вывод, удаление, хранение, поиск, сортировка и печать следующей информации:

- сотрудниках фирмы и их должностях;
- клиентах;
- поставщиках;
- товарах и их видах;
- продажах;
- движениях товаров по складам;
- анализ заказов в разрезе сотрудников и товаров.

Входной информацией системы является бухгалтерская и регистрационная информация, представленная ниже.

Бухгалтерская информация:

- информация о стоимости всех товаров.

Регистрационная информация:

- информация о клиенте;
- информация о сотрудниках;
- информация о должностях сотрудников;
- информация о поставщиках;
- информация о складах;
- информация о товарах;
- классификация товаров;
- информация о видах движения товаров по складам;
- складские остатки.

Выходной информацией системы являются отчеты и аналитические диаграммы. Минимальный перечень формируемых в системе отчетов следующий:

- список сотрудников, должностей, поставщиков, клиентов, товаров;
- список товаров на складах;
- отчет об остатках на складах;
- отчет о движениях товаров;
- дополнительные совмещенные списки.

1.5 Общая характеристика организации решения задачи на ЭВМ

Для организации работы пользователей сети с информационными ресурсами, распределенными по различным компьютерам, необходимы три составляющих:

– программа, установленная на компьютере пользователя, которая может осуществлять сетевой запрос с целью получения объекта, и предназначенная для его обработки (например, просмотра, изменения или печати документа);

– программа, установленная, как правило, на компьютере, где расположен информационный объект, которая может осуществлять по запросу поиск и пересылку объекта, а также упорядочивание доступа к нему нескольких пользователей;

– правила (протокол) взаимодействия между этими программами.

Технология взаимодействия, в которой одна программа запрашивает выполнение какой-либо совокупности действий, а другая ее выполняет, называется технологией «клиент-сервер». Участники такого взаимодействия называются соответственно клиентом и сервером. Достаточно часто клиентом (или сервером) называют компьютеры, на которых функционирует то или иное клиентское (или серверное) программное обеспечение [14].

При использовании клиент-серверной технологии доступа к базам данных обработка запроса одного пользователя представлен на рисунке 1. Обращение к базе данных происходит посредством SQL-запроса [15].



Рисунок 1.3 – Схема технологии клиент-сервер

При необходимости произвести обработку информации, хранящейся в базе данных, запущенное на компьютере пользователя клиентское приложение, работающее с базой данных, формирует запрос на языке SQL. Сервер базы данных принимает запрос и обрабатывает его самостоятельно. Никакой массив данных (файл) по сети не передается. После обработки запроса на компьютер пользователя передается только результат [16].

В серьезных клиент-серверных СУБД существуют дополнительные механизмы, снижающие нагрузку на сеть, снижающие требования к пользовательским компьютерам [17].

Разработка информационной системы обычно подразделяется на 2 этапа:

- разработка базы данных;
- разработка программного обеспечения (для оформления информационной системы и управления базой данных) [18].

Одной из важнейших и значимых характеристик базы данных является их целостность [19].

Под целостностью базы данных в общем случае понимается ее готовность к работе. Целостность базы данных – сложное понятие, имеющее много аспектов, таких как:

- физическая целостность, то есть сохранность информации на магнитных носителях и корректность форматов данных;
- логическая целостность, под которой понимается непротиворечивость данных в базе;
- актуальность данных, т. е. соответствие данных реальному положению вещей [20].

Потеря целостности базы данных может произойти от сбоев аппаратуры ЭВМ, ошибок в программном обеспечении, неверной технологии ввода и корректировки данных, низкой достоверности самих данных, и т. д.

Поэтому обеспечить целостность базы реального объема весьма сложно, но это необходимая мера её работоспособности [21].

Таким образом, автоматизированная система должна:

- проводить контроль вводимой информации;
- блокировать некорректные действия пользователя при работе с системой;
- обеспечивать целостность данных.

Поскольку реляционная модель базы данных получила очень широкое распространение, так как основывается на математических принципах из теории множеств и логики, то именно этот тип модели был выбран для реализации базы данных проекта.

Реляционная модель – это совокупность данных, состоящих из набора двумерных таблиц. Реляционная форма является удобной и наиболее привычной формой представления данных [22].

Выбор методик разработки приложений становится основной задачей в условиях стремительного роста рынка. Разработка концепции Rapid Application Development (RAD) стало основой для создания гибкой, адаптивной системы разработки приложений. RAD – модель быстрой разработки приложений, ключевыми для которой является скорость и удобство программирования [23].

Использование этой технологии предполагает, что разработка проводится очень ограниченной командой разработчиков и за короткий срок, благодаря использованию инкрементного прототипирования с применением инструментальных средств визуального моделирования и разработки. Технология RAD предусматривает активное привлечение заказчика на всех этапах проектирования [24].

1.6 Анализ существующих разработок и обоснование выбора технологии проектирования

Одной из важнейших задач любого разработчика является подбор наиболее оптимальной среды разработки. Для этого необходимо провести анализ рынка программного обеспечения разработчика прикладных программ.

«1С:Управление торговлей 8» – это современный инструмент для повышения эффективности бизнеса торгового предприятия [25].

«1С:Управление торговлей 8» позволяет в комплексе автоматизировать задачи оперативного и управленческого учета, анализа и планирования торговых операций, обеспечивая тем самым эффективное управление современным торговым предприятием.

«1С:Управление торговлей 8» автоматизирует следующие направления хозяйственной деятельности:

- управление отношениями с клиентами;
- управление процессами продаж;
- управление запасами;
- управление закупками;
- управление складом;
- управление финансами;
- контроль и анализ целевых показателей деятельности предприятия

[26].

В программе могут регистрироваться как уже совершенные, так и еще только планируемые хозяйственные операции. «1С:Управление торговлей 8» автоматизирует оформление практически всех первичных документов торгового и складского учета, а также документов движения денежных средств.

«1С:Управление торговлей 8» рассчитана на любые виды торговых операций. Реализованы функции учета – от ведения справочников и ввода первичных документов до получения различных аналитических отчетов.

Одним из основных недостатков данного программного обеспечения – это дорогостоящее и очень сложное внедрение. Настройка, внедрение и запуск должны осуществляться квалифицированным 1С-специалистом. Кроме того, часто возникает необходимость решать возникшие вопросы по работе в программе оперативно. Для этого также придется заказывать услуги поддержки 1С, которые на сегодняшний момент обладают достаточно высокой стоимостью [27].

Последнее время все большую популярность набирают облачные хранилища данных. Интернет-сервис для управления торговлей «МойСклад» предназначен для автоматизации малого и среднего бизнеса. Позволяет управлять продажами и закупками, контролировать взаиморасчеты, работать с клиентской базой, вести складской учет, а также печатать все необходимые для ведения бизнеса документы.

Небольшие компании и частные предприниматели, используя сервис «МойСклад», получают возможность автоматизации оперативного (управленческого) учета без лишних расходов и потерь времени. Немаловажным фактом является поддержка программных продуктов с открытым кодом (например, ОС Linux и OpenOffice.org) [28].

Более крупные компании на базе сервиса смогут объединить в одной системе несколько филиалов, складов, офисов или точек продаж, организовать работу со своей дистрибьюторской сетью, а также адаптировать «МойСклад» под свои специфические требования.

Основные преимущества данного продукта:

- интуитивно понятный интерфейс и логика работы;
- не требуется установка на компьютер или сервер;
- пробный период и консультации по настройке бесплатно;
- легко обучить сотрудников работе с системой.

Основными недостатками является достаточно высокая стоимость продукта, которая зависит от выделяемого объема памяти на сервере, а также проблемы с периодическим отсутствием доступа к серверу, что может сильно сказаться на работе компании.

Еще одним достаточно популярным программным продуктом на рынке является «Subtota» [29]. Простая и удобная система учета розницы и склада, экономящая время и повышающая доходность бизнеса. «Subtotal» – это облачная система, которая обеспечивает автоматизацию работы розничного магазина или сети магазинов. Позволяет вести полноценный учет склада, учет продаж. Инструменты контроля и развития бизнеса: продажи по каждому магазину, оптимизация остатков по складу, сумма денег в кассе, средняя сумма чека, маржинальность и чистая прибыль.

Плюсы системы:

- интеграция с бухгалтерским и пакетом программ 1С: Предприятие;
- возможность работы в сервисе с планшетов и ноутбуков;
- наличие приложения для создания шаблонов составных товаров;
- дружелюбный интерфейс.

Минусы системы:

- отсутствие некоторых специализированных функций;
- частые сбои в работе программы;
- сложность интеграции с другими программными продуктами.

Таким образом, программное обеспечение для такого класса задач как автоматизация деятельности торгового предприятия довольно широко распространено, но все программные продукты не заточены под бизнес-процессы отдельно взятого предприятия, в силу того что в торговой деятельности начиная от торговли продуктами питания и заканчивая торговлей бытовой техникой сильно различается специфика работы [30].

Поэтому для того что бы получить такую информационную систему предприятию желательно разработать или доработать существующую

систему максимально подходящую для бизнеса, учтя все нюансы деятельности фирмы.

Выводы по первому разделу: первый раздел содержит технико-экономическую характеристику компьютерного сервиса «RossComService». Кроме того, в нем описаны результаты проведения существующих разработок подобных систем, а также торговая деятельность компьютерного сервиса «RossComService», построена функциональная модель «КАК ЕСТЬ».

2 Обоснование проектных решений автоматизации

2.1 Обоснование проектных решений по техническому обеспечению

Техническое обеспечение – совокупность технических средств, необходимых для функционирования информационной системы [31].

К техническому обеспечению относят:

- комплекс технических средств;
- организационные формы использования;
- инструкции по эксплуатации техники.

К комплексу технологических средств был предъявлен следующий набор требований:

- высокая производительность;
- надежность;
- безопасность;
- эффективность функционирования
- рациональное распределение по уровням обработки.

При этом должен соблюдаться принцип экономичности выбора и использования технических средств, их эксплуатацию. В таблице 2.1 представлены характеристики оптимальной для обеспечения торговой деятельности компании ЭВМ.

Таблица 2.1 – Характеристика технического обеспечения ЭВМ

Наименование	Характеристика
Материнская плата	ASUS H81M-R LGA 1150, Intel H81, 2xDDR3-1600 МГц, 1xPCI-Ex16, аудио 7.1, Micro-ATX
Процессор	LGA 1150 Intel H81 1600 МГц
Жесткие диски	2xDDR3
Видеокарты	ASUS Nvidia Geforce gtx 460 SE 1 gb
Звуковая карта	C-Media 8738 4-channel PCI(oem)

Продолжение таблицы 2.1

Наименование	Характеристика
Приводы CD-R/RW	CD-RW+DVD Teac52*32*52*16 DW-552G(oem)
Модули памяти. Flash-накопители	USB Flash Drive (14 Гб)
Корпуса	AeroColler Strike X
Модемы	ZyXEL Keenetic Extra II
Клавиатура	Defender OfficeMate SM-820
Мышь оптическая.	DEXP CM-503BU
Сетевой фильтр	Сетевой фильтр (5 розеток) PilotGL 2м
Монитор	19.5" LG 20MP48A-P [20MP48A-P.ARUZ]
Сетевой адаптер	Casio AD-E95100LG
Принтер	Pantum P2200

2.2 Обоснование проектных решений по информационному обеспечению

Информационное обеспечение включает в себя организацию информационной базы. Разработчики определяют связи между информационными потоками, формируют пакет входной документации в электронном виде, определяют использование нормативно-справочной документации и уровни доступа к информации пользователей различных категорий в зависимости от сущности решаемых ими задач [32].

Информационное обеспечение включает массивы данных, средства их описания, сбора, хранения и выдачи, которые должны в совокупности создать наилучшие условия для централизованной интегрированной обработки информации, повысить надёжность и достоверность получаемой информации.

Выходные потоки данных представлены в виде списков клиентов, сотрудников, поставщиков, товаров, отчетах о складских остатках и движении товаров на складе, а также о продажах в разрезе товаров с возможностью графического представления показателей [33].

Управление всеми бизнес-процесса происходит на основе приказ и распоряжений должностных лиц компании, а также ряда законов и внутренних кадровых документов.

Компанией были предоставлены ряд справочников, таких как «Склад», «Вид движения», «Должности» и «Вид товара», каждый из которых должен иметь возможность расширения по требованию заказчика.

Делегирование полномочий – это процесс передачи части функций руководителя другим управляющим или сотрудникам для достижения конкретных целей организации. Используется для улучшения и оптимизации рабочей силы руководителя.

На рисунке 2.1 представлены возможности доступа к каждой планируемой таблицы базы данных на уровне должностей (ролей), имеющихся в компании.

	Клиенты	Сотрудники	Должности	Поставщики	Товары	Виды товаров	Склады	Реестр движения товаров	Покупки
Директор	+	+	+	+	+	+	+	+	+
Администратор магазина	+	+	+	+	+	+	+/-	+/-	+
Продавец-консультант	+	-	-	+/-	+/-	+	-	+/-	+
Кладовщик	-	-	-	+	+	+	+	+	-
Администратор системы	+	+	+	+	+	+	+	+	+

Рисунок 2.1 – Делегирование полномочий

2.3 Обоснование проектных решений по программному обеспечению

Программное обеспечение – это совокупность программ, позволяющих осуществить на компьютере автоматизированную обработку информации [34].

Программное обеспечение ЭВМ делят на следующие группы:

- прикладные программы, непосредственно обеспечивающие выполнение необходимых пользователям работ;
- системные программы, выполняющие различные вспомогательные функции;
- инструментальные программные системы, облегчающие процесс создания новых программ для компьютера.

В компании есть реализованный модуль «Учет сервисных услуг». В основе него также лежит «клиент-серверная» технология. Для разработки новой подсистемы магазин компьютерной техники и её эксплуатации использовался следующий комплекс программного обеспечения:

- операционная система Windows 7;
- СУБД Firebird;
- утилита IVExpert;
- средство разработки приложения C++ Builder.

Операционная система Windows 7 является одной из самых распространенная в настоящее время операционная система. Windows 7 имеет высочайший уровень устойчивости к ошибочным и неосторожным действиям пользователей.

Перечень настроек Windows 7 весьма богат – начиная от графики и заканчивая правами доступа и настройкой безопасности. Несомненный плюс операционной системы состоит в том, что она совместима с современными распространенными программами [35].

Firebird – кроссплатформенная реляционная система управления базами данных (РСУБД), работающая на macOS, Linux, Microsoft Windows и Unix платформах [35].

Firebird является сервером баз данных. Один сервер Firebird может обрабатывать несколько сотен независимых баз данных, каждую с множеством пользовательских соединений. Он является полностью свободным от лицензионных отчислений даже для коммерческого использования [35].

Firebird позволяет работать со всеми основными объектами баз данных: таблицы, ракурсы, триггеры, генераторы, хранимыми процедура и событиями.

Утилита IBExpert предназначена для разработки и администрирования баз данных InterBase и Firebird, а также для выбора и изменения данных, хранящихся в базах.

Она обладает множеством облегчающих работу компонентов: визуальный редактор для всех объектов базы данных, редактор SQL и исполнитель скриптов, отладчик для хранимых процедур и триггеров, построитель области, инструмент для импорта данных из различных источников, собственный скриптовый язык, а также дизайнер баз данных.

Кроме того, распространяется бесплатно для всех стран постсоветского пространства.

C++ Builder был выбран для реализации win-приложения, так как это продукт, который наиболее эффективно осуществляет доступ приложения к базам данных.

C++ Builder – одна из ведущих сред разработки для создания Internet-приложений, настольных и распределенных приложений, а также приложений, основанных на модели «клиент-сервер». Он сочетает в себе простоту среды быстрой разработки приложений, или RAD-среды, с мощностью и производительностью языка C++, совместимого со стандартом ANSI.

Язык программирования C++ является одним из самых распространённых. Он применяется для разработки самых разных приложений – от сложных многоуровневых бизнес-систем до высокопроизводительных программ визуализации данных и систем реального времени [36].

Достоинства языка C++:

- масштабируемость : на языке C++ разрабатывают программы для самых различных платформ и систем;
- возможность работы на низком уровне с памятью, адресами, портами;
- возможность создания обобщенных алгоритмов для разных типов данных, их специализация, и вычисления на этапе компиляции, используя шаблоны.

2.4 Обоснование выбора программных средств

Существует ряд программных продуктов позволяющих осуществлять разработку RAD-приложений.

Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств [36].

Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода,

так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

Главным недостатком для заказчика является её высокая стоимость, а для разработчика – необходимость большого объема знаний и опыта для разработки, внедрения и в будущем интеграции с разработанной ранее системой.

CodeLite – открытая кроссплатформенная среда разработки программного обеспечения для языков программирования C/C++, PHP и Node.js, использующая инструментарий wxWidgets [36].

Особенности CodeLite: управление проектами, автодополнение, рефакторинг кода, подсветка синтаксиса, интеграция Subversion и Git, интеграция Cscope, интеграция UnitTest++, интерактивный отладчик, настроенный над GDB, и мощный редактор исходного кода (на основе Scintilla).

CodeLite распространяется по лицензии GNU General Public License v2 или более поздней версии. Является свободным программным обеспечением.

Одним из основных недостатков этой платформы разработки является её не стабильность и множество мелких недочетов в управления сборкой проекта, который достаточно сильно увеличивают время разработки приложения [36].

Компания «RossComService» имеет действующую лицензию на использование программного средства C++ Builder, что является основным ключом для выбора среды разработки. Это связано с тем, что чаще всего, большая часть бюджета проекта используется на покупку прав на использование необходимо программного обеспечения для разработки и дальнейшей эксплуатации системы. В данном случае рациональнее всего использовать имеющее программное средство.

Кроме того, основной модуль «Учет сервисных услуг» реализован в этой среде разработки, что поможет качественно провести интеграцию систем и наладить их взаимодействие.

Выводы по второму разделу: данный раздел посвящен обоснованию проектных решений автоматизации, включающий в себя техническое, информационное и программное обеспечение, а также обоснование выбора среды разработки.

3 Проектная часть

3.1 Информационная модель и ее описание

После оценки эффективности выполняемых действий и выявления слабых сторон существующей системы на основе модели «КАК ЕСТЬ» строится функциональная модель «КАК ДОЛЖНО БЫТЬ».

На рисунке 3.1 видно, что одним из основных механизмов теперь выступает автоматизированная система компьютерного магазина. Кроме того, к должностным инструкциям сотрудников добавилась инструкция по эксплуатации автоматизированной системы, в которой подробно описаны возможности и основные действия всех должностных лиц компании.

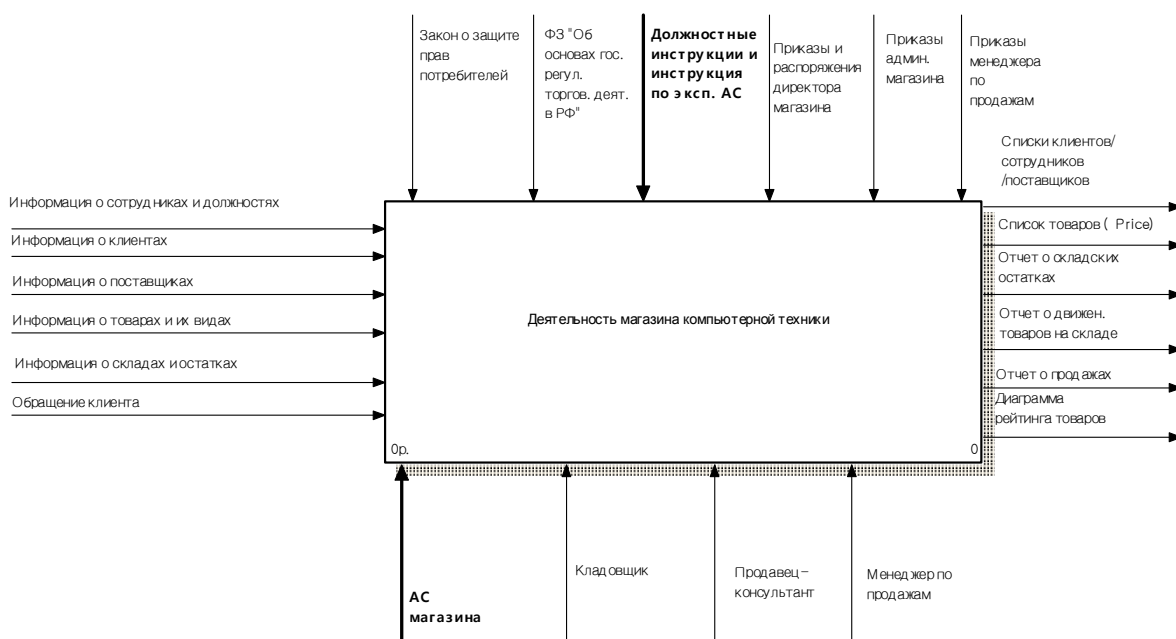


Рисунок 3.1 – Контекстная диаграмма деятельности магазина «КАК ДОЛЖНО БЫТЬ»

Все основные бизнес-процессы деятельности магазина остаются неизменными (рисунок 3.2).

В процессе закупки товаров отсутствует роль кладовщика, который был задействован в процессе анализа складских остатков. С использованием автоматизированной системы менеджер по продажам сможет получать данную информацию самостоятельно из системы. Кроме того, блок «Формирования отчетности» был исключен, так как система будет формировать их автоматически.

В процессе складского учета теперь формируется реестр движения товаров, на основе которых формируется вся необходимая отчетность. Реестр формируется системой в автоматическом режиме после добавления кладовщиком проводки.

Продажа компьютерной техники теперь также осуществляется с помощью автоматизированной системы, а формирование отчетности также проходит без участия продавца-консультанта.

Менеджер по продажам полностью отстраняет от проведения анализа торговой деятельности, так как формирование отчетности теперь полностью переложено на автоматизированную систему.

Подробные диаграммы бизнес-процессов закупки товара, складского учета и продаж техники представлены на рисунках Б.1 – Б.3.

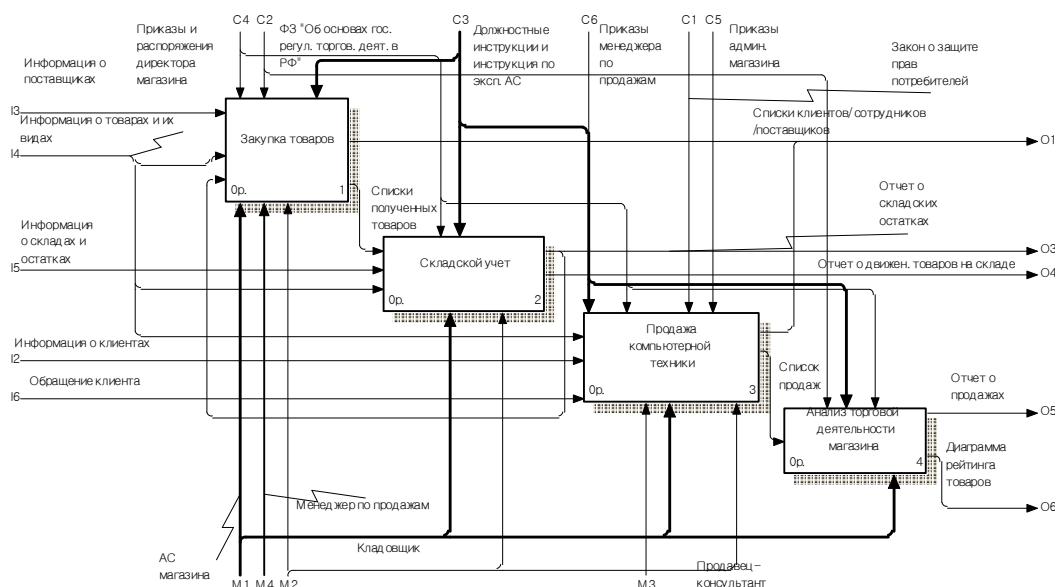


Рисунок 3.2 – Диаграмма-декомпозиция деятельности магазина «КАК ДОЛЖНО БЫТЬ»

Таким образом, функциональная диаграмма «КАК ДОЛЖНО БЫТЬ» показала, что все сотрудники магазина будут освобождены от рутинной работы формирования отчетности, что даст возможность обслуживать большее число клиентов. Кроме того, некоторые должностные обязанности сотрудников были ликвидированы.

3.2 Разработка базы данных

Проектирование баз данных – процесс создания схемы базы данных и определения необходимых ограничений целостности [37].

Проектирование БД является очень важным этапом, от которого зависят последующие этапы разработки СУБД. Время, затраченное разработчиком на проектирование БД, обычно окупается высокой скоростью реализации проекта.

Перед созданием базы данных необходимо располагать описанием выбранной предметной области, которое должно охватывать реальные объекты и процессы, иметь всю необходимую информацию для удовлетворения предполагаемых запросов пользователя и определить потребности в обработке данных.

На основе такого описания на этапе проектирования базы данных осуществляется определение состава и структуры данных предметной области. Структура данных предметной области может отображаться информационно-логической моделью. На основе этой модели легко создается реляционная база данных.

Поскольку реляционная модель базы данных получила очень широкое распространение, так как основывается на математических принципах из теории множеств и логики, то именно этот тип модели был выбран для реализации базы данных проекта.

Реляционная модель – это совокупность данных, состоящих из набора двумерных таблиц. Реляционная форма является удобной и наиболее привычной формой представления данных [38].

База данных для организации деятельности компьютерного магазина состоит из 11 сущностей, представленных в таблице 3.1.

Таблица 3.1 – Сущности базы данных и их атрибуты

Наименование сущности	Атрибуты	Тип данных	Ключ
1. Люди	Номер лица	Integer	PK
	Фамилия	Varchar	
	Имя	Varchar	
	Отчество	Varchar	
	Телефон	Varchar	
	Адрес	Varchar	
	Дата рождения	Date	
	Паспортные данные	Varchar	
2. Должности	Номер должности	Integer	PK
	Должность	Varchar	
3. Сотрудники	Номер сотрудника	Integer	PK
	Номер лица	Integer	
	Номер должности	Integer	FK
4. Клиенты	Номер клиента	Integer	
	Номер лица	Integer	
5. Поставщик	Номер поставщика	Integer	PK
	Наименование организации	Varchar	
	Адрес	Varchar	
	Телефон	Varchar	
6. Вид товара	Расчетный счет	Varchar	
	Номер вида	Integer	PK
	Наименование	Varchar	
7. Товары	Номер товара	Integer	PK
	Наименование товара	Varchar	
	Номер вида товара	Integer	FK
	Номер поставщик	Integer	FK
	Цена	Float	
8. Склады	Номер склада	Integer	PK
	Наименование склада	Varchar	
9. Вид движения	Номер движение	Integer	PK
	Наименование	Varchar	
10. Складские проводки	Номер проводки	Integer	PK
	Вид движения	Integer	FK

Продолжение таблицы 3.1

Наименование сущности	Атрибуты	Тип данных	Ключ
10. Складские проводки	Номер сотрудника	Integer	FK
	Номер товара	Integer	FK
	Номер склада	Integer	FK
	Количество	Integer	
	Дата проводки	Date	
11. Продажи	Номер продажи	Integer	PK
	Номер сотрудника	Integer	FK
	Номер клиента	Integer	FK
	Номер товара	Integer	FK
	Количество	Integer	
	Стоимость продажи	Float	
	Дата проводки	Date	
	Статус продажи	Varchar	

Следующим шагом является отображение в среду спецификаций инфологической модели предметной области. Эту стадию называют логическим (дatalogическим) проектированием базы данных. Ее результатом является концептуальная схема базы данных, включающая определение всех информационных единиц и связей, в том числе задание типов, характеристик и имен.

Дatalogическое проектирование приводит к разработке схемы базы данных, то есть совокупности схем отношений, которые адекватно моделируют абстрактные объекты предметной области и семантические связи между этими объектами. Необходимым условием грамотной организации базы данных является преобразование отношений базы данных к виду, отвечающему нормальным формам [35].

Нормализация предназначена для приведения структуры базы данных к виду, обеспечивающему минимальную логическую избыточность, и не имеет целью уменьшение или увеличение производительности работы или же уменьшение или увеличение физического объема базы данных.

В результате приведения модели к 3 нормальной форме получаем следующую модель:

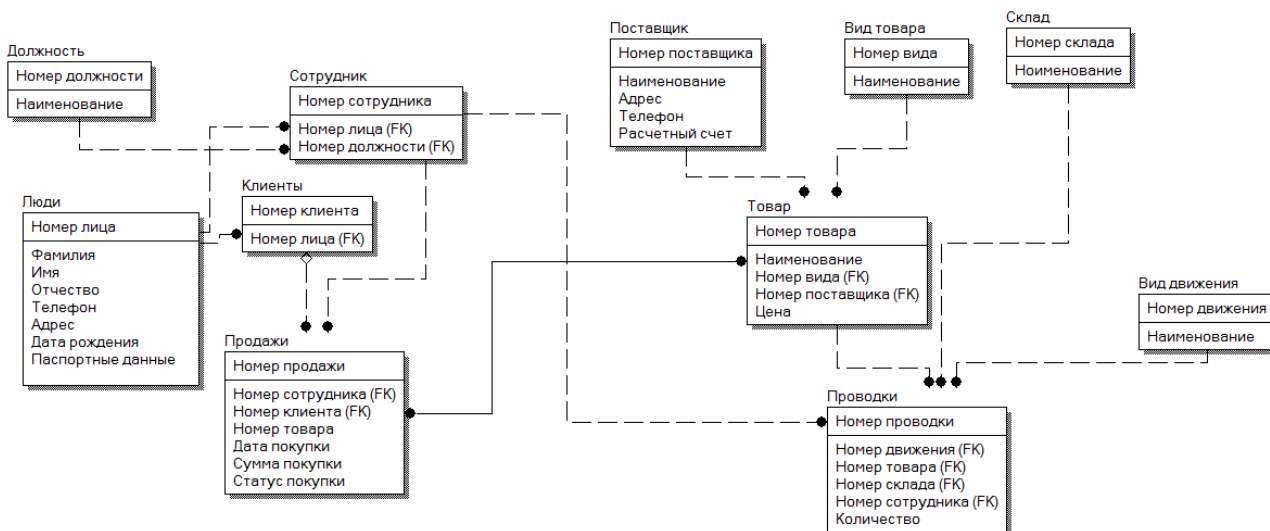


Рисунок 3.3 – Логическая модель базы данных

Следующим шагом необходимо разработать физическую модель базы данных.

Физическая модель базы данных определяет способы размещения данных в среде хранения и способы доступа к этим данным, которые поддерживаются на физическом уровне (рисунок 3.4).

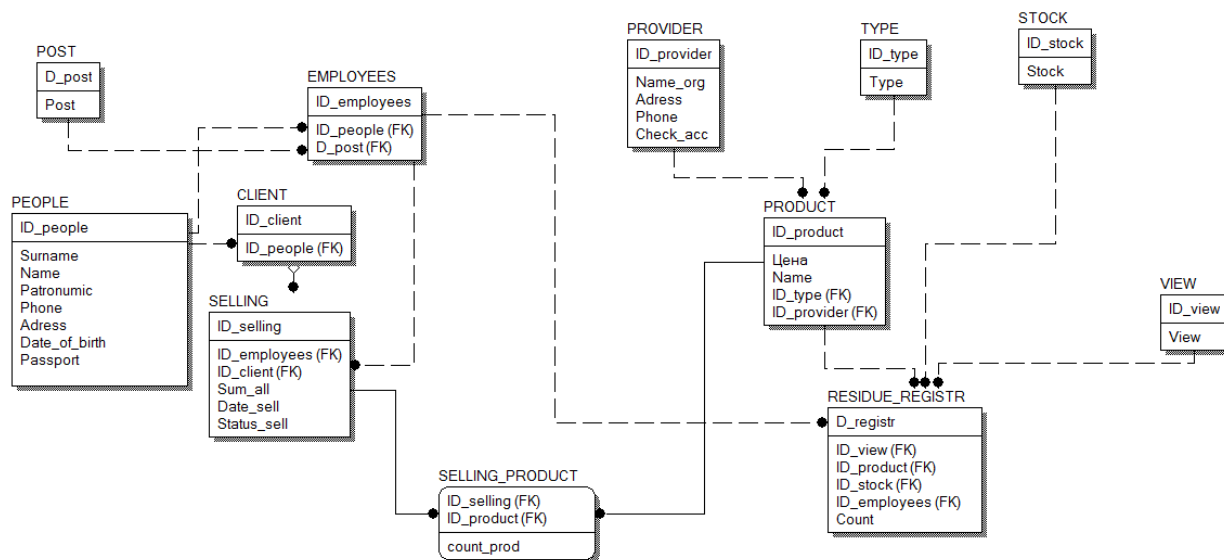


Рисунок 3.4 – Физическая модель базы данных

В логической модели базы данных существует связь «многие-ко-многим» между сущностями «Покупка» и «Товар». Данный вид связи не

используется в реляционной базе данных. Поэтому в физической модели данная связь представлено с помощью выделения дополнительной таблицы «Продажа-товар», которая будет содержать информацию обо всех товара в каждой покупке.

Для построения моделей использовалось AllFusion ERwin Data Modeler. Это CASE-средство для проектирования и документирования баз данных, которое позволяет создавать, документировать и сопровождать базы данных, хранилища и витрины данных [34].

Для разработки базы данных была выбрана утилита IVExpert, предназначенная для разработки и администрирования баз данных InterBase и Firebird, а также для выбора и изменения данных, хранящихся в базе данных.

База данных была создана и зарегистрирована в утилите под именем «comp_shop.fdb». Далее в ней был создан ряд необходимых объектов, таких как домены, генераторы и триггеры для автоматического ведения ключевых полей таблиц (рисунок 3.5).

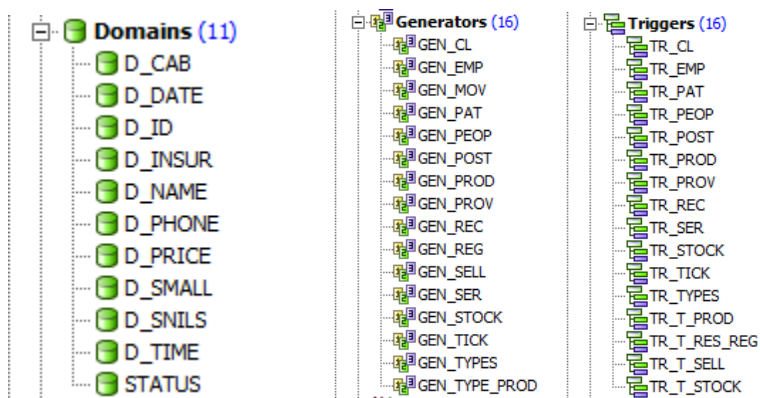


Рисунок 3.5 – Созданные домены

Далее были созданы все необходимые таблицы (рисунок 3.6) и просмотры. Просмотры являются временными наборами данных, существующими только во время работы с ними.

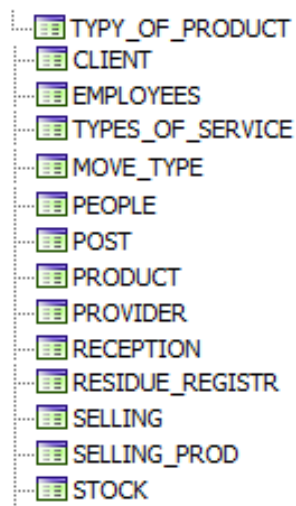


Рисунок 3.6 – Созданные таблицы

Кроме того, для каждой таблицы было создано по 3 хранимые процедуры, которые позволяют удалять, добавлять и изменять данные.

Исходные коды создания генераторов, триггеров, таблиц, просмотров и хранимых процедур находятся в приложении В.

3.3 Разработка пользовательского интерфейса автоматизированной подсистемы

Пользовательский интерфейс – это программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером. Основу такого взаимодействия составляют диалоги [32].

Разработка пользовательского начинается с построения модели дерева функций, выполняемых системой.

Функция – это процесс, операция или действие, выполняющиеся над объектом для достижения определенных целей [35].

На рисунке 3.7 представлена иерархия функций управления и обработки данных, которая состоит из 2 основных групп – основные и служебные.

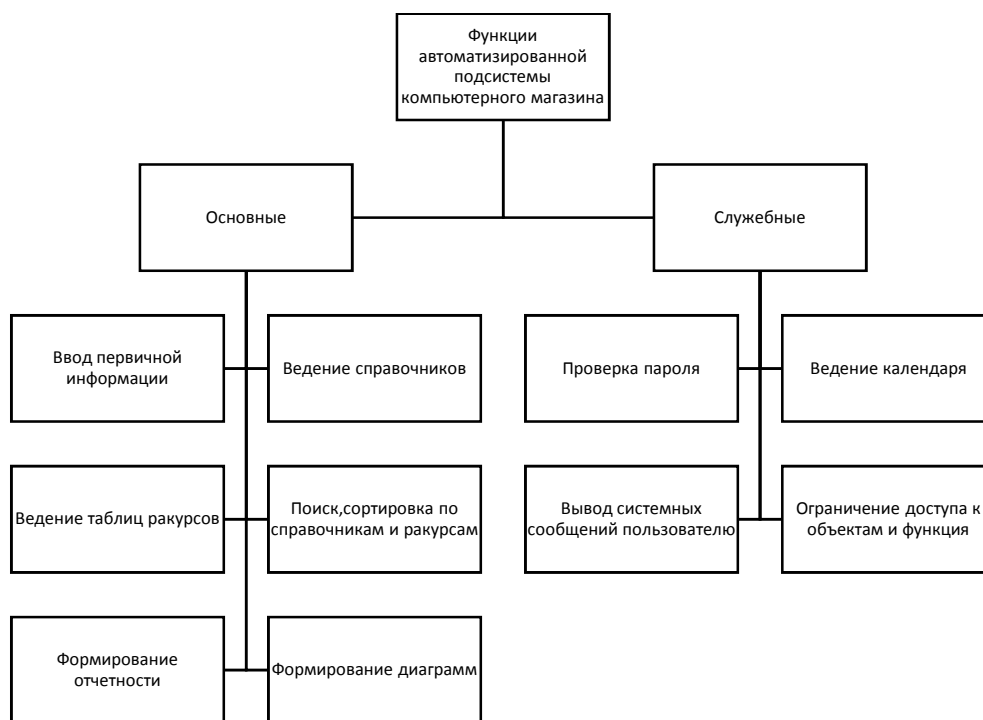


Рисунок 3.7 – Дерево функций систем

На основе дерева функций была разработана схема диалога (рисунок 3.8).

Под диалогом понимают регламентированный обмен информацией между человеком и компьютером, осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи [37].

Каждый диалог состоит из отдельных процессов ввода/вывода, которые физически обеспечивают связь пользователя и компьютера. Обмен информацией осуществляется передачей сообщения.

Внутри каждого из процессов выполняется ряд основных функций, а также переход к любому другому процессу, на который есть полномочия у выбранной при входе в систему роли, а также выход из системы.

К основным функциям относится функция добавления, удаления, изменения, поиска, сортировки, печати отчетов, также ряд дополнительных для формирования проводок оприходование, списание, перемещение. При осуществлении первых 3-х функций возможен вызов справочников, которые помогут правильно заполнить форму ввода.

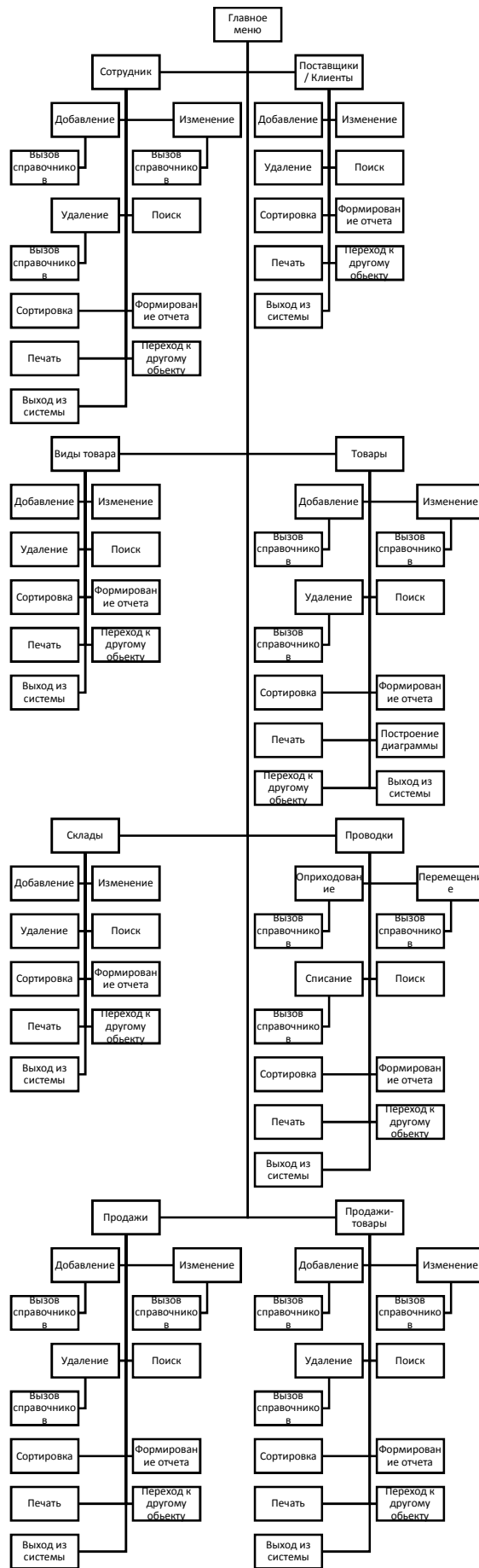


Рисунок 3.8 – Схема сценария диалога системы

С учетом всех описанных выше схем был разработан пользовательский интерфейс. При запуске системы появляется форма входа в систему, представленная на рисунке 3.9. Роли в системе созданы с делегированием полномочий в зависимости от должности. Форма обладает выпадающими списками справочников должностей и сотрудников компании.

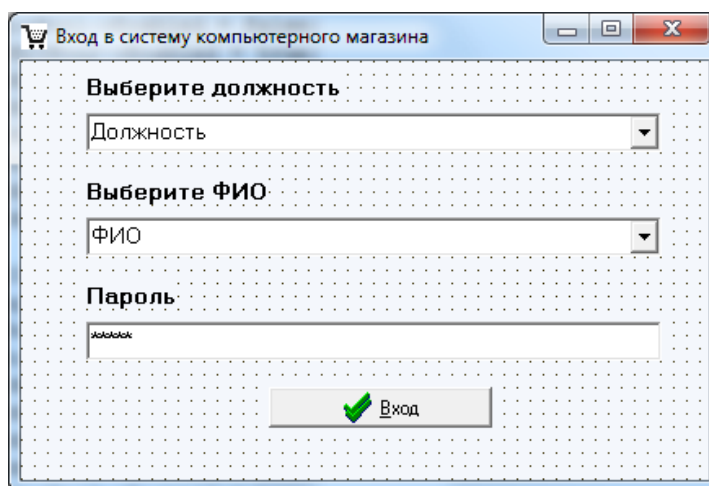


Рисунок 3.9 – Форма входа в систему

Далее был разработан интерфейс главного меню системы (рисунок 3.10).

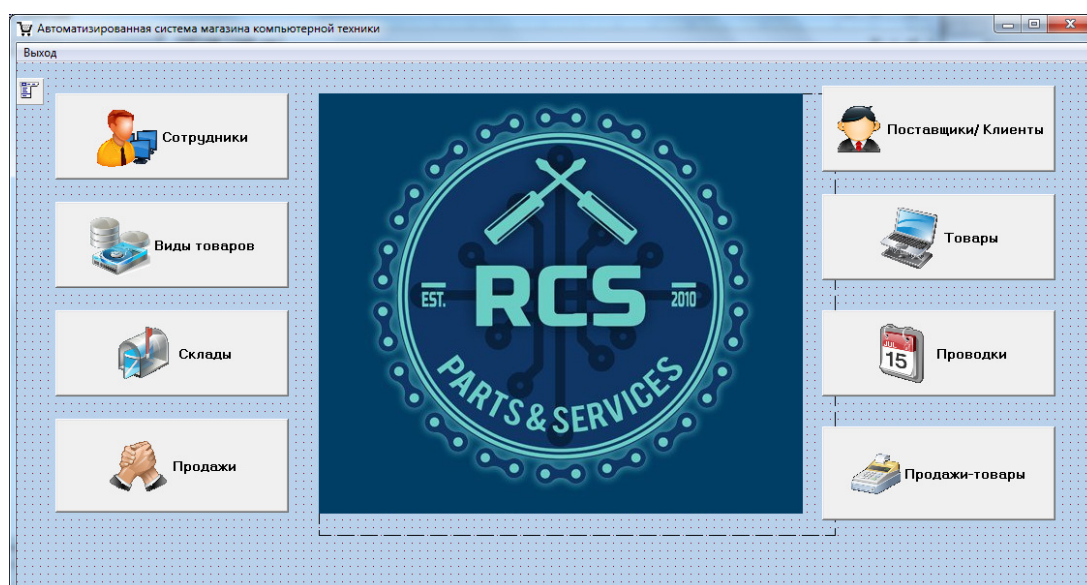


Рисунок 3.10 – Главное меню системы

Для всей системы был выбран синий цвет как основной. Это связано с несколькими факторами. Во-первых, основное меню содержит корпоративную эмблему в оттенках синего цвета. Во-вторых, этот цвет в психологии ассоциируется с покоем и сосредоточенностью, что очень важно для пользователей системы. Разработанный интерфейс достаточно прост и интуитивно понятен.

Далее был разработан интерфейс формы для работы с объектами базы данных. На рисунке 3.11 представлена форма для работы с таблицей «Сотрудники», которая соответствует сценарию диалога.

Номер сотрудника	Фамилия	Имя	Отчество	Должность	Телефон
1	Сергеева	Анна	Ивановна	Директор	89206571261
2	Смирнов	Илья	Игоревич	Администратор магазина	89257493688
3	Иванова	Ирина	Сергеевна	Менеджер по продажам	8904369852
4	Белобров	Сергей	Александрович	Продавец-консультант	8962478521
5	Рыжков	Иван	Александрович	Продавец-консультант	8742369145
6	Борисова	Екатерина	Анатольевна	Кладовщик	8526364128

Рисунок 3.11 – Основная форма «Сотрудники»

Она состоит из основного окна вывода информации, полей для ввода данных, панелей для сортировки и поиска, панели меню работы с таблицей и основным меню перехода между формами приложения. Было принято решение использовать данный дизайн для остальных форм приложения.

Кроме того, была разработана отдельная форма для вывода всех справочников системы (рисунок 3.12).

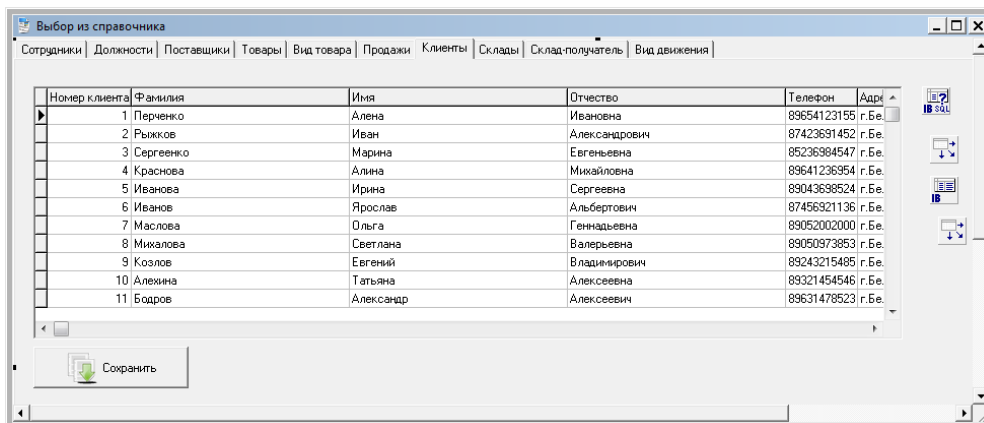


Рисунок 3.12 – Форма для отображения справочников

3.4 Описание контрольного примера реализации проекта

Автоматизированная система компьютерного магазина позволяет вести следующие бизнес-процессы:

- ведение справочников;
- учет товаров на складе;
- учет продаж.

Безопасность информационной системы обеспечивает авторизация. Авторизация – это предоставление определённому лицу или группе лиц прав на выполнение определённых действий; а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий [39].

Осуществим вход под ролью системного администратора, так как он обладает полномочиями на все объекты системы (рисунок 3.13).

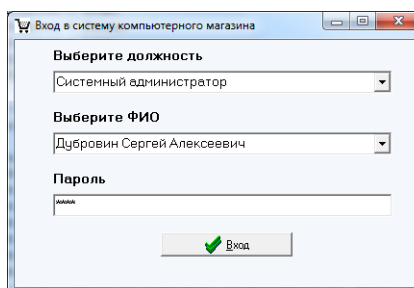


Рисунок 3.13 – Авторизация пользователя в системе

При введении верного пароля откроется основное меню системы, представленное на рисунке 3.14.

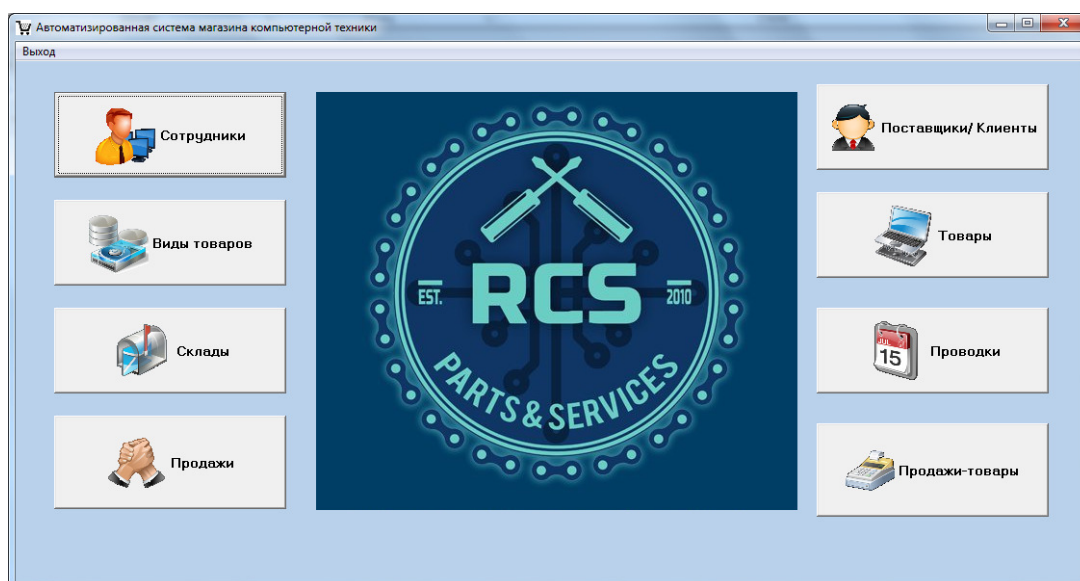


Рисунок 3.14 – Основное меню системы для роли системного администратора

Если пароль был введен неверно, то пользователь получит соответствующее системное сообщение (рисунок 3.15).

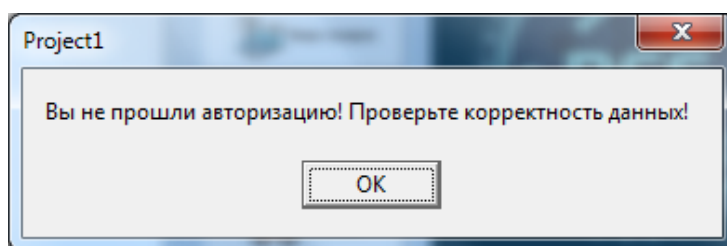


Рисунок 3.15 – Системное сообщение об ошибке авторизации

Для пользователя с ограниченными полномочиями на объекты системы в основном меню будут доступны только разрешенные объекты. На рисунке 3.16 представлено основное меню пользователя с ролью «Продавец-консультант».



Рисунок 3.16 – Основное меню для роли «Продавец-консультант»

Из роли системного администратора перейдем к форме таблицы «Сотрудники», нажав по соответствующей кнопке меню (рисунок 3.17).

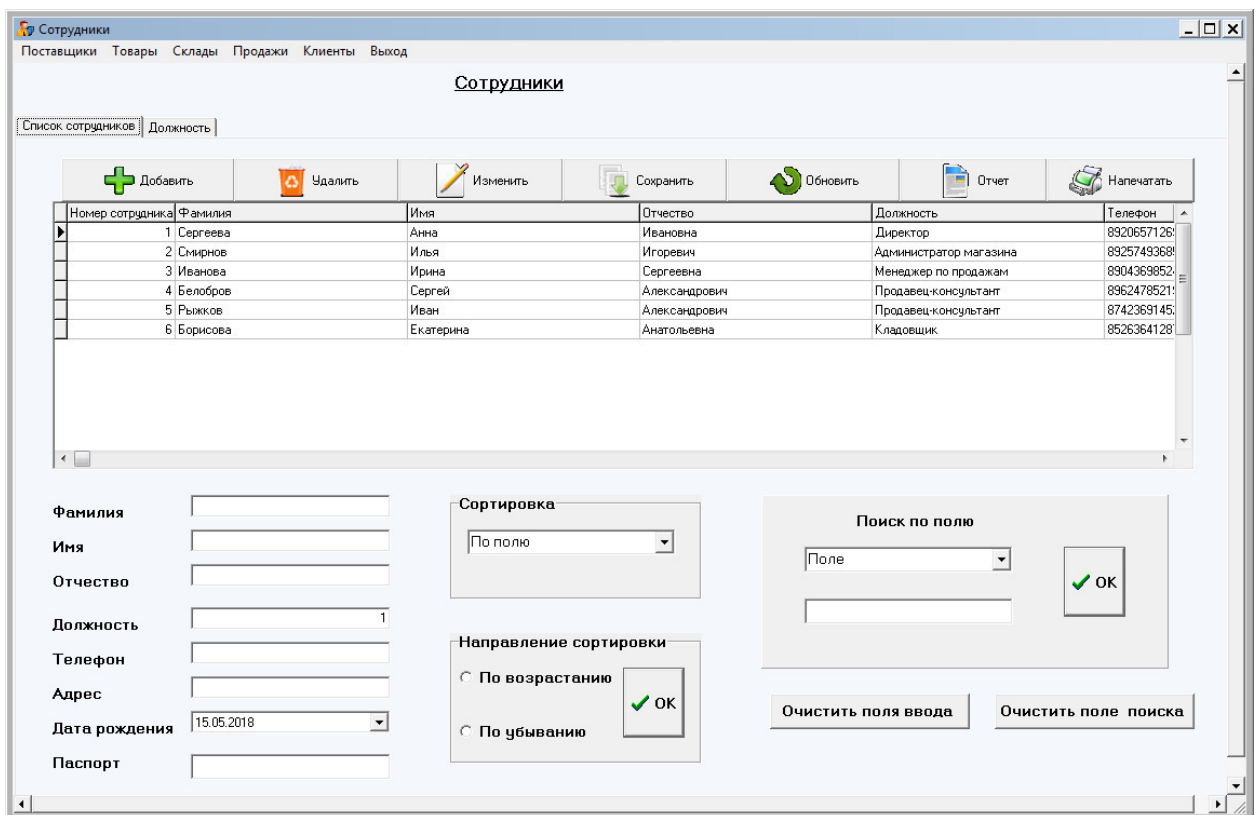


Рисунок 3.17 – Форма таблицы «Сотрудники»

Далее были выполнены основные действия пользователя при работе с данной таблицей. Одной из важнейших функций является функция добавления записи, то есть введение в систему данных о новом сотруднике. Для этого необходимо заполнить все поля.

Поле «Дата рождения» может быть указано вручную в стандартном формате день, месяц, год, а также с помощью выбора даты в появляющемся календаре (рисунок 3.18).

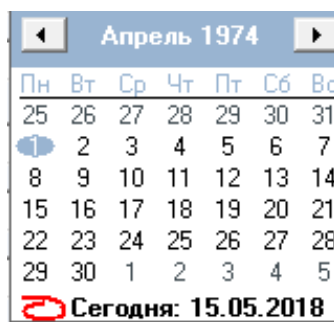


Рисунок 3.18 – Календарь выбора даты

При указании должности сотрудника всплывает окно со справочников, из которого необходимо выбрать необходимое значение и нажать кнопку «Сохранить» (рисунок 3.19).

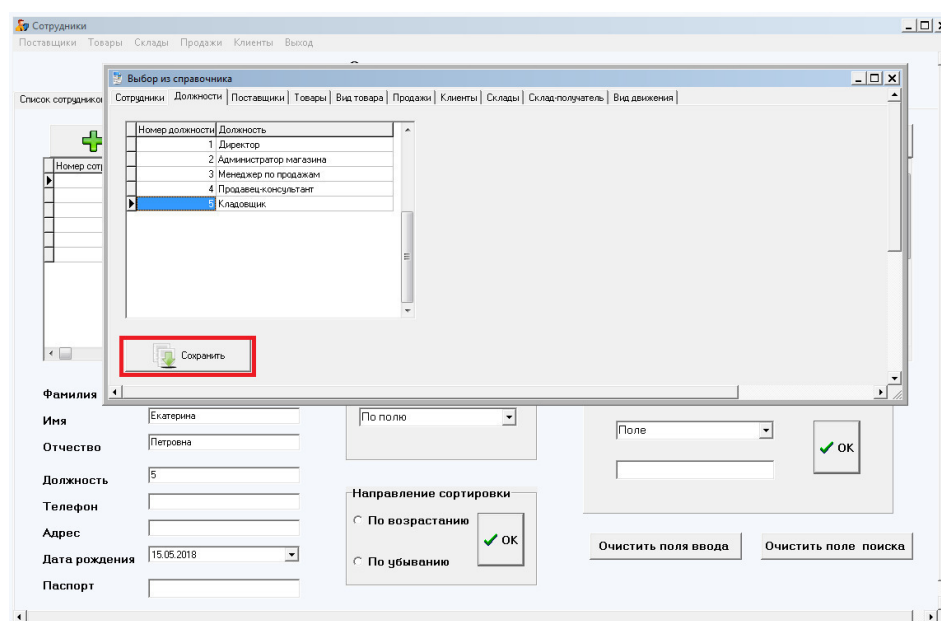


Рисунок 3.19 – Справочник должностей

После заполнения полей необходимо нажать кнопку «Добавить» (рисунок 3.20).

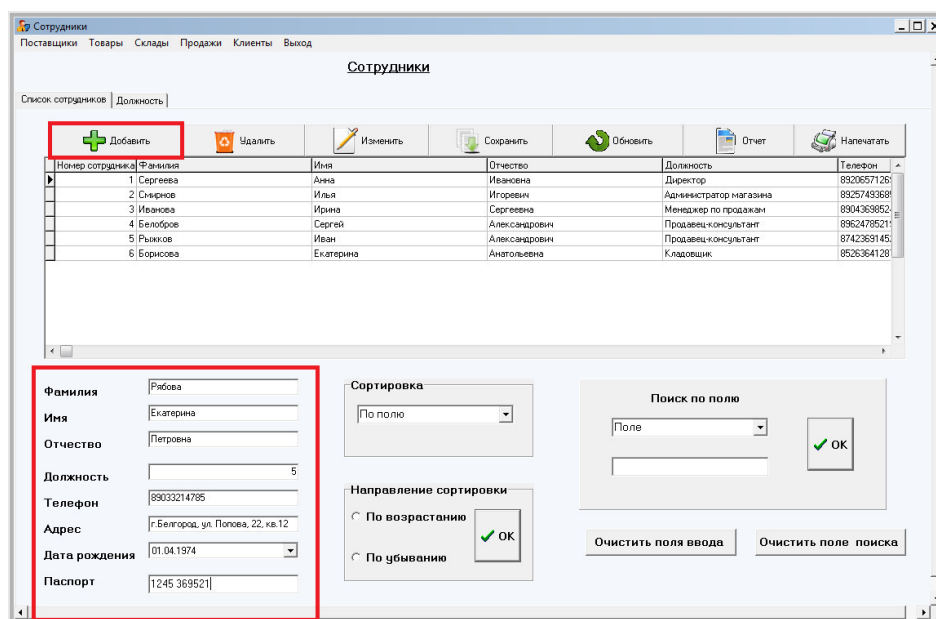


Рисунок 3.20 – Добавление нового сотрудника

На рисунке 3.21 представлен результат добавления нового сотрудника, который получил автоматически номер «9».

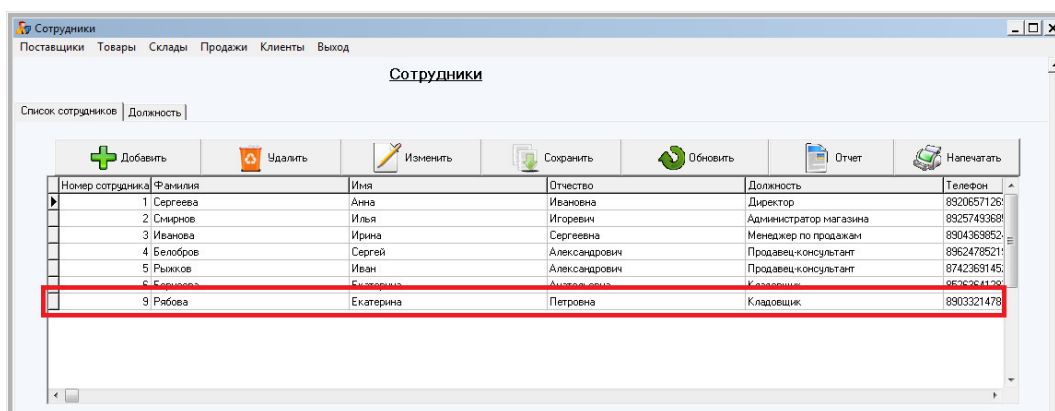


Рисунок 3.21 – Отображение в списке нового сотрудника

Если пользователем были заполнены не все поля, он получит сообщение, которое изображено на рисунке 3.22.

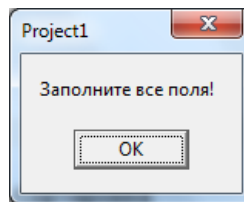


Рисунок 3.22 – Системное сообщение при добавлении записи

Для того, чтобы изменить одну из записей необходимо выбрать ее с помощью курсора в представленном списке, затем нажать кнопку «Изменить», после чего появится предупреждение о переходе в режим редактирования (рисунок 3.23).

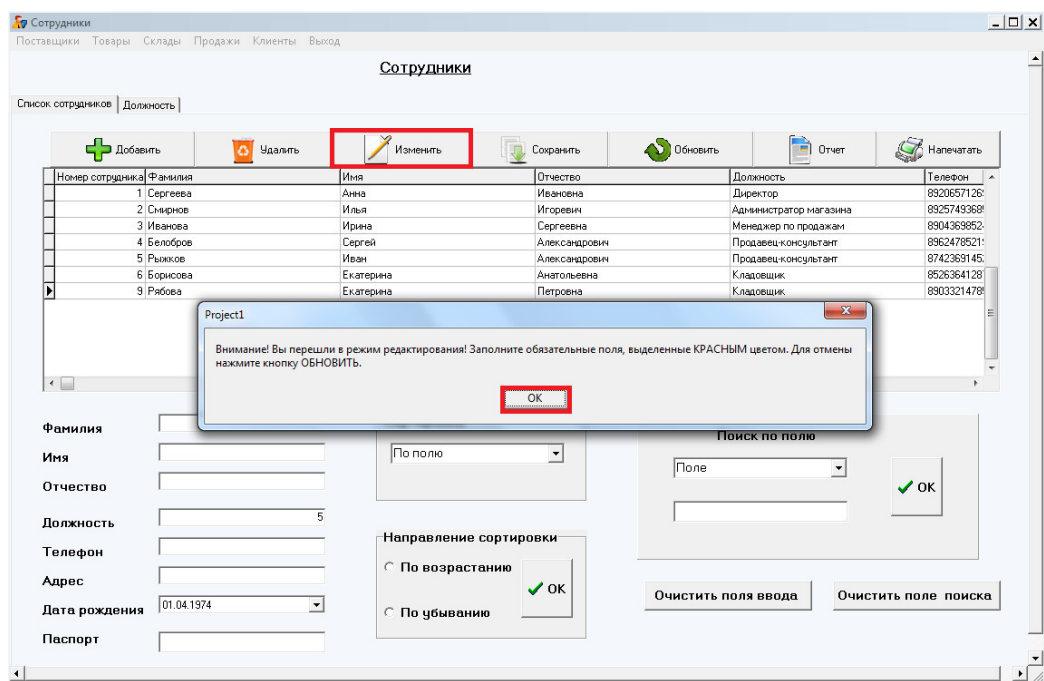


Рисунок 3.23 – Переход в режим редактирования

После чего, красным выделяются те поля, которые обязательны для ввода, если будут не указаны остальные поля, они сохранят текущие значения. Пользователю достаточно ввести только те, которые необходимо изменить.

После ввода необходимо нажать кнопку «Сохранить». Все введенные изменения сразу будут отображены в выведенном списке сотрудников (рисунок 3.24).

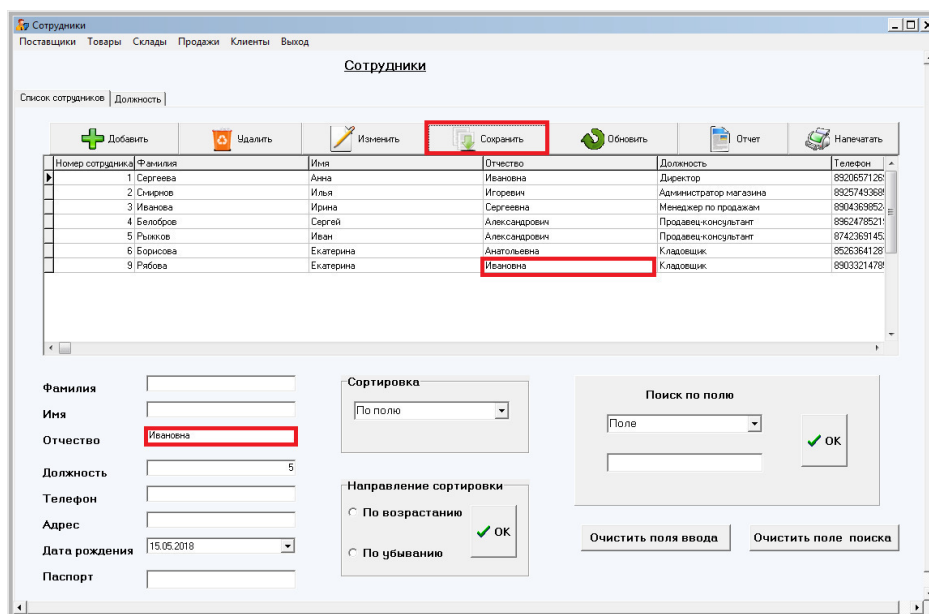


Рисунок 3.25 – Изменение отчества сотрудника

Для удаления сотрудника из списка необходимо выделить его курсором в списке и нажать кнопку «Обновить» (рисунок 3.26).

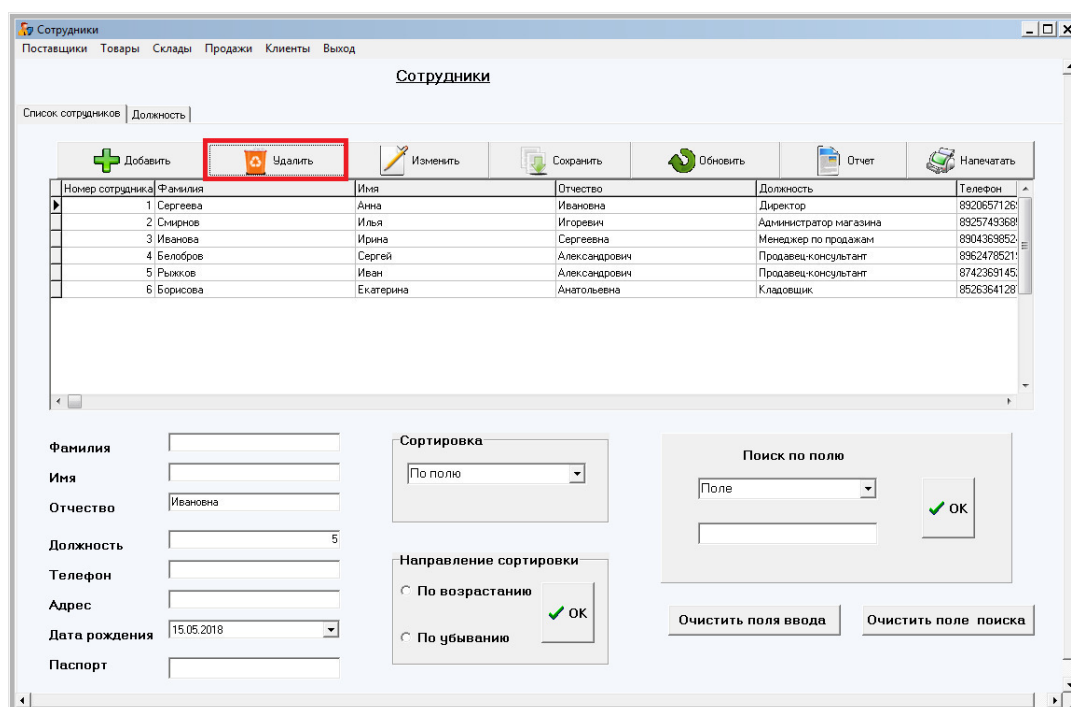
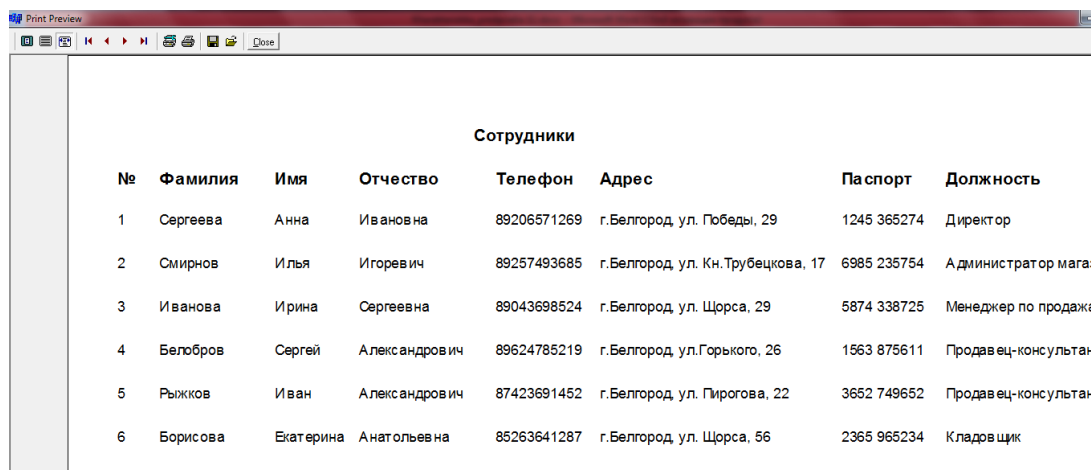


Рисунок 3.26 – Удаление добавленного сотрудника

Для того, чтобы просмотреть отчет по списку сотрудников необходимо нажать кнопку «Отчет» (рисунок 3.27). Отчет можно распечатать, нажав соответствующую кнопку.

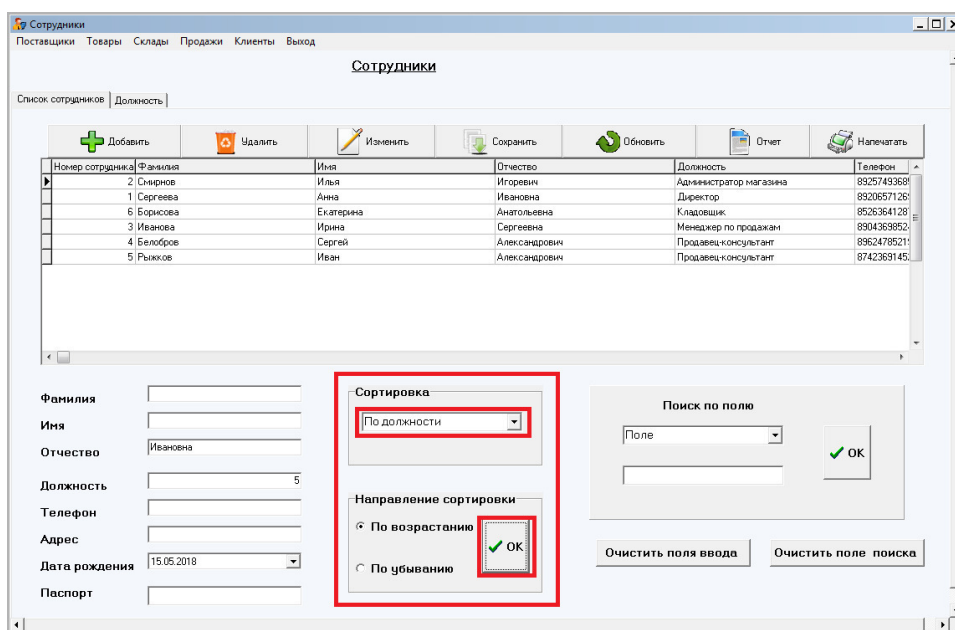


The screenshot shows a 'Print Preview' window with a table titled 'Сотрудники'. The table has 8 columns: №, Фамилия, Имя, Отчество, Телефон, Адрес, Паспорт, and Должность. There are 6 rows of employee data.

№	Фамилия	Имя	Отчество	Телефон	Адрес	Паспорт	Должность
1	Сергеева	Анна	Ивановна	89206571269	г.Белгород, ул. Победы, 29	1245 365274	Директор
2	Смирнов	Илья	Игоревич	89257493685	г.Белгород, ул. Кн.Трубецкого, 17	6985 235754	Администратор магази
3	Иванова	Ирина	Сергеевна	89043698524	г.Белгород, ул. Щорса, 29	5874 338725	Менеджер по продажам
4	Белобров	Сергей	Александрович	89624785219	г.Белгород, ул.Горького, 26	1563 875611	Продавец-консультант
5	Рыжков	Иван	Александрович	87423691452	г.Белгород, ул. Пирогова, 22	3652 749652	Продавец-консультант
6	Борисова	Екатерина	Анатольевна	85263641287	г.Белгород, ул. Щорса, 56	2365 965234	Кладовщик

Рисунок 3.27 – Просмотр отчета перед печатью

Система позволяет также осуществлять сортировку списка по ряду полей. Для этого нужно выбрать из выпадающего списка поле для сортировки и направление, после чего нажать кнопку «ОК» (рисунок 3.28).



The screenshot shows the 'Сотрудники' application window. At the top, there is a menu bar with 'Поставщики', 'Товары', 'Склады', 'Продажи', 'Клиенты', and 'Выход'. Below the menu is a toolbar with buttons: 'Добавить', 'Удалить', 'Изменить', 'Сохранить', 'Обновить', 'Отчет', and 'Напечатать'. The main area contains a table of employees. Below the table, there are input fields for 'Фамилия', 'Имя', 'Отчество', 'Должность', 'Телефон', 'Адрес', 'Дата рождения', and 'Паспорт'. A 'Сортировка' section is highlighted with a red box, containing a dropdown menu set to 'По должности' and a radio button selected for 'По возрастанию'. An 'OK' button is next to the radio button. To the right, there is a 'Поиск по полю' section with a dropdown menu and an 'OK' button. At the bottom, there are buttons for 'Очистить поля ввода' and 'Очистить поле поиска'.

Рисунок 3.28 – Сортировка сотрудников по должности

Также система дает возможность вести поиск по полю. Для этого необходимо выбрать из выпадающего списка название поля и ввести текст поиска (рисунок 3.29). После нажатия на кнопку «ОК» в списке останутся только те записи, которые удовлетворяют условию поиска.

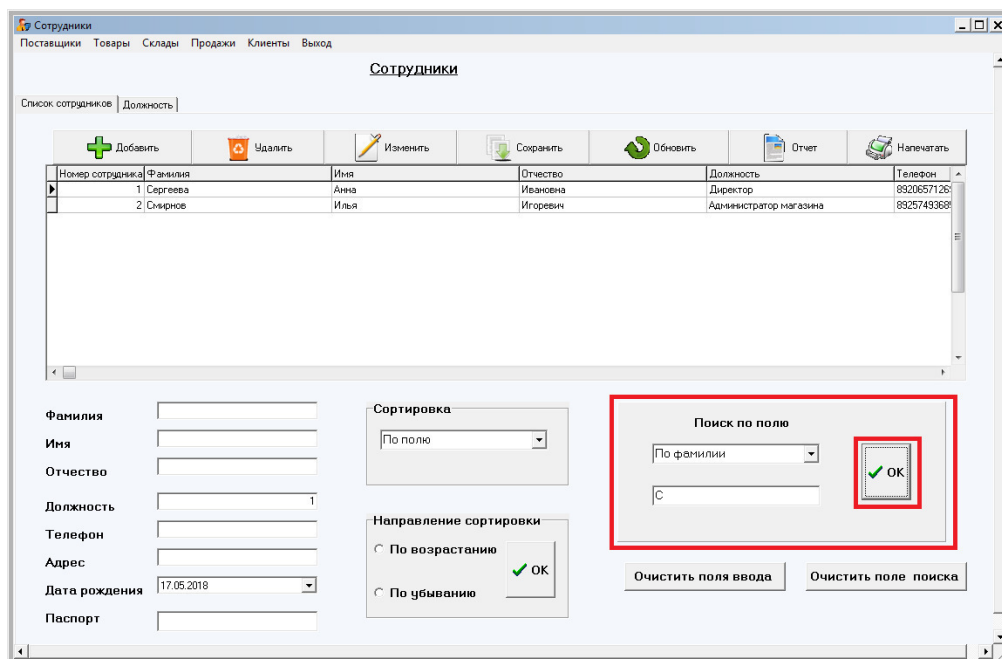


Рисунок 3.29 – Сортировка сотрудников по должности

Если по заданному тексту поиска ничего не найдено, пользователь получит следующее системное представление на рисунке 3.30.

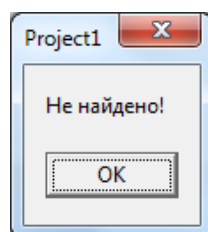


Рисунок 3.30 – Системное сообщение

Существует ряд дополнительных кнопок, которые позволяют очищать поля ввода и поиска, а также обновлять представленные записи таблицы (без фильтров) и печать отчет. Кроме того, с помощью панели меню есть

возможность перейти в любой доступный раздел и осуществить выход из системы (рисунок 3.31).

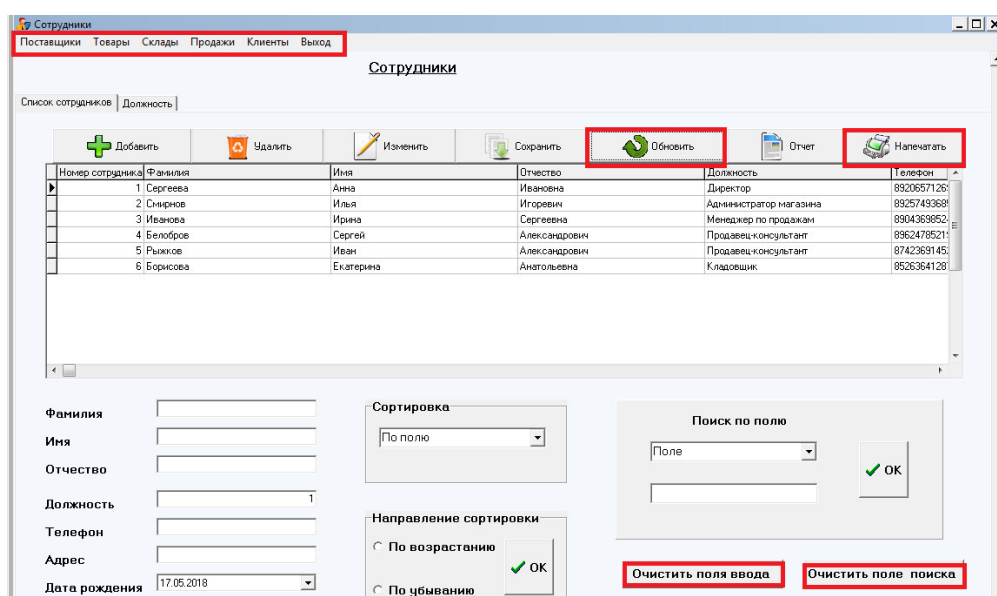


Рисунок 3.31 – Дополнительные кнопки

Одним из важнейших бизнес-процессов является учет товаров на складе. Этот процесс необходимо вести в пункте меню «Проводки». Интерфейс представлен на рисунке 3.32.

Для того чтобы оприходовать товар на склад, необходимо заполнить все доступные поля для ввода: вид движения, товар, склад-получатель, количество, сотрудник и дата проводки. Далее нажать кнопку «Оприходовать» и в списке появится новая проводка. Подобным же образом проводится списание товара со склада.

Перемещение товара выполняется следующим образом: в первую очередь необходимо нажать кнопку «Переместить», после чего пользователь получает сообщение системы и поле «Склад-отправитель» станет доступен для выбора из справочника. После указания всех полей и нажатия на кнопку «Сохранить» будет сформировано 2 проводки: списание со склада-отправителя и оприходование на склад получатель (рисунок 3.33).

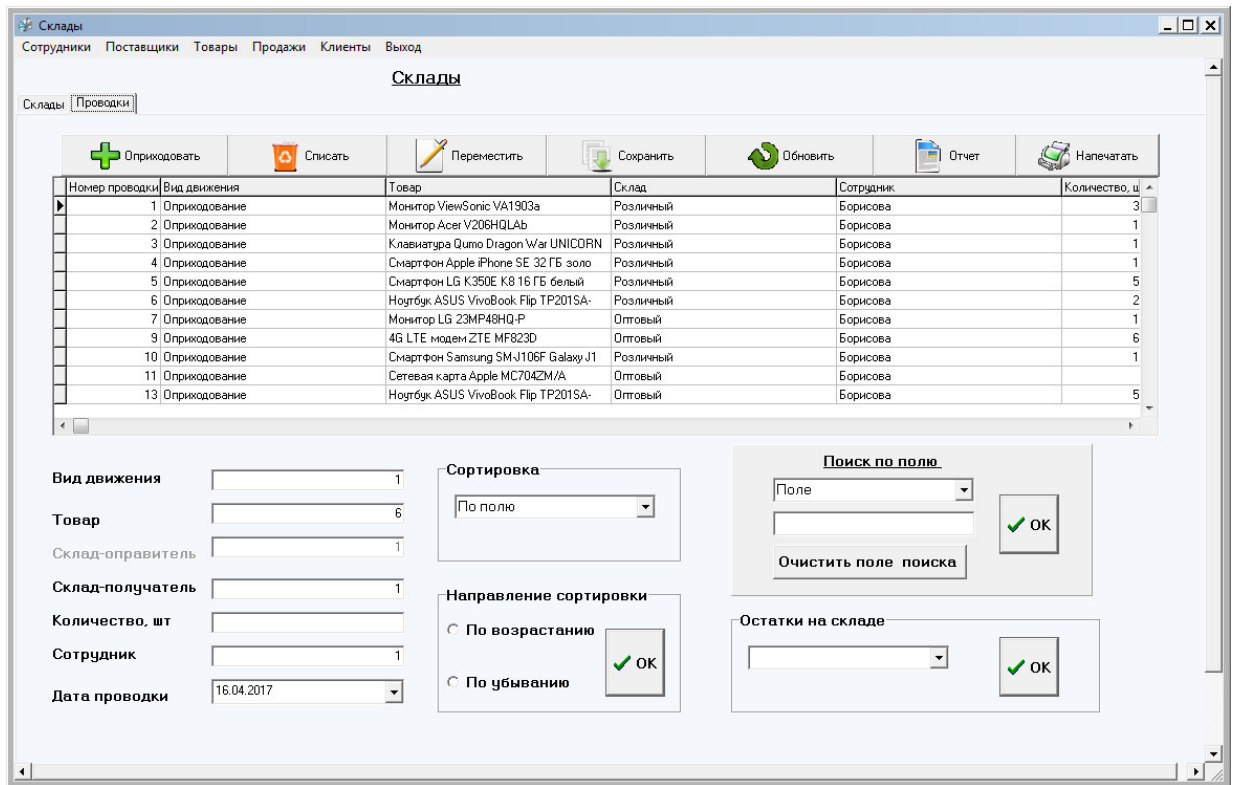


Рисунок 3.32 – Интерфейс формы «Проводки»

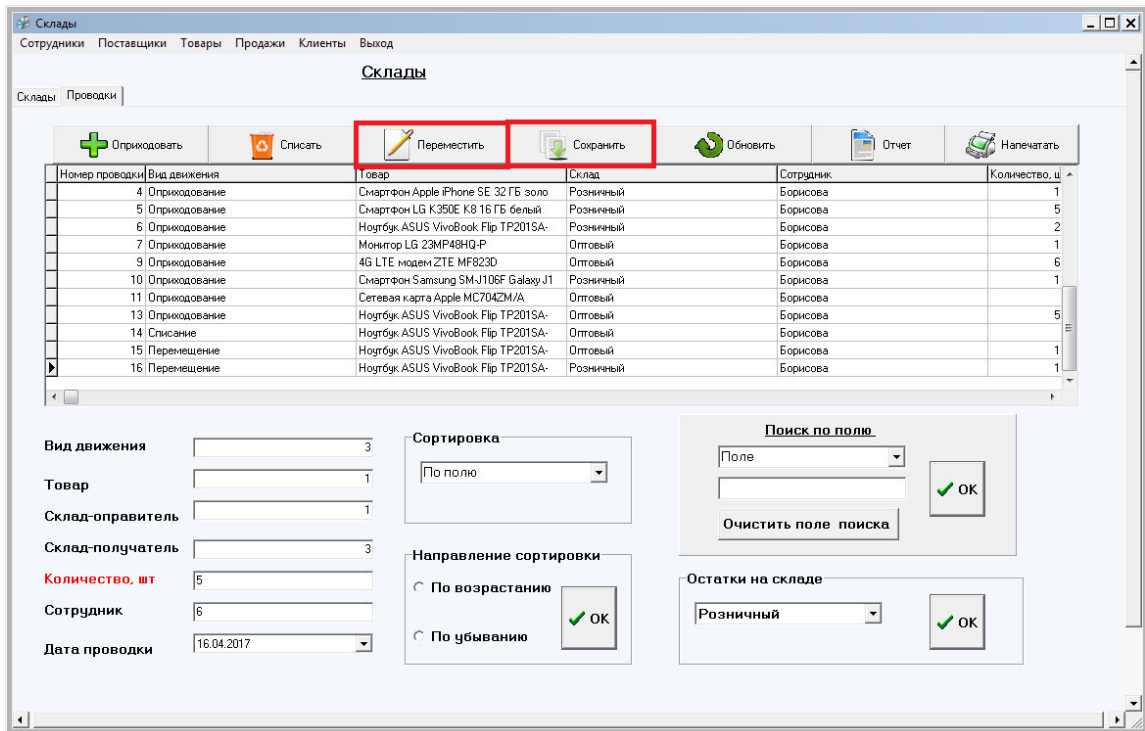


Рисунок 3.33 – Перемещение товара

Кроме того, после успешного добавления проводки пользователь также получит системное сообщение, представленное на рисунке 3.34.

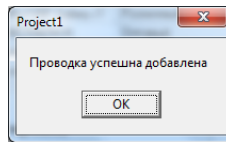


Рисунок 3.34 – Системное сообщение

Кроме стандартных функций данная форма позволяет пользователю посмотреть остатки на каждом из складов и создает автоматически отчетную форму (рисунок 3.35-3.36).

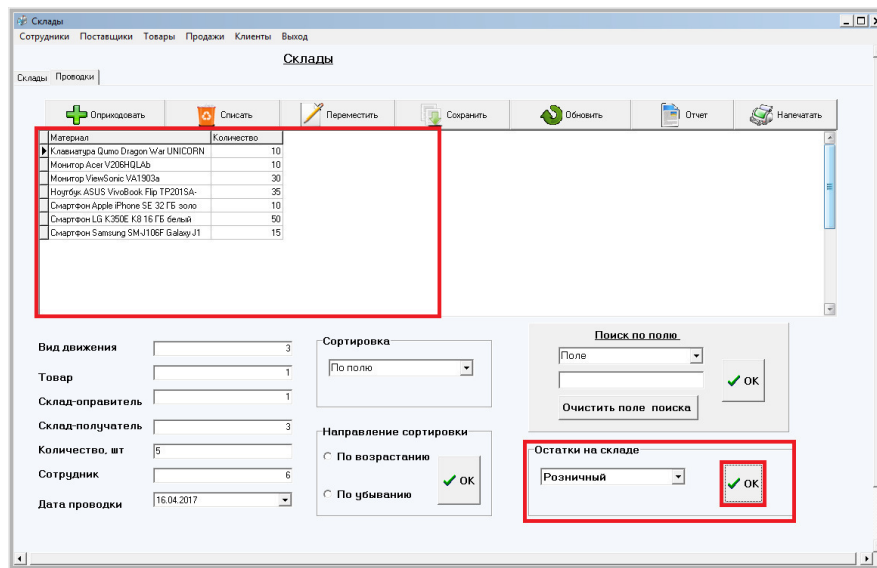


Рисунок 3.35 – Просмотр остатков на складе

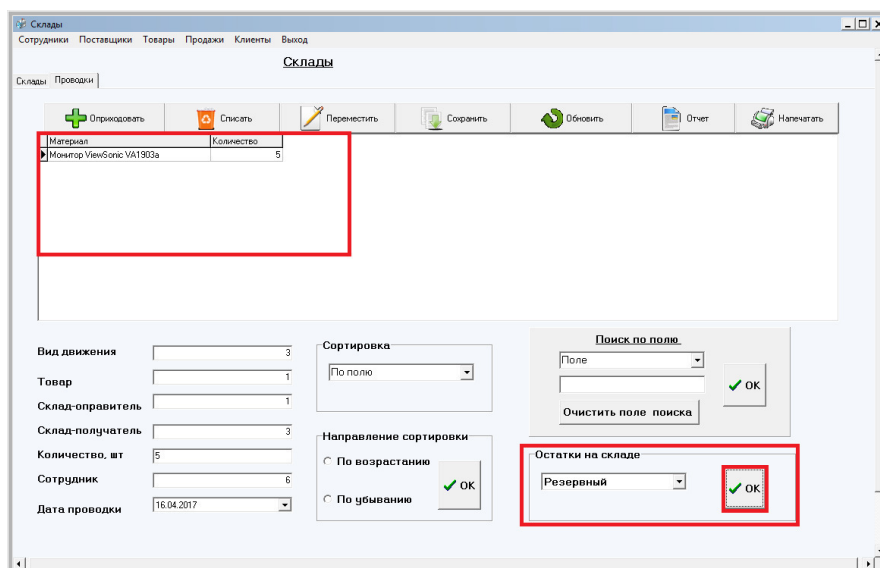


Рисунок 3.36 – Просмотр остатков на складе

Учет покупок пользователя ведется в разделе меню «Покупки». Этот процесс состоит из 2 этапов. В первую очередь необходимо создать покупку, заполнив поля ввода (или выбрать значение из справочника), представленные на рисунке 3.37. После нажатия кнопки «Добавить» покупка появится в выведенном списке.

Очень важно указать при добавлении товара верный статус заказа «Принят». С помощью статусов покупок легко проводить анализ продаж.

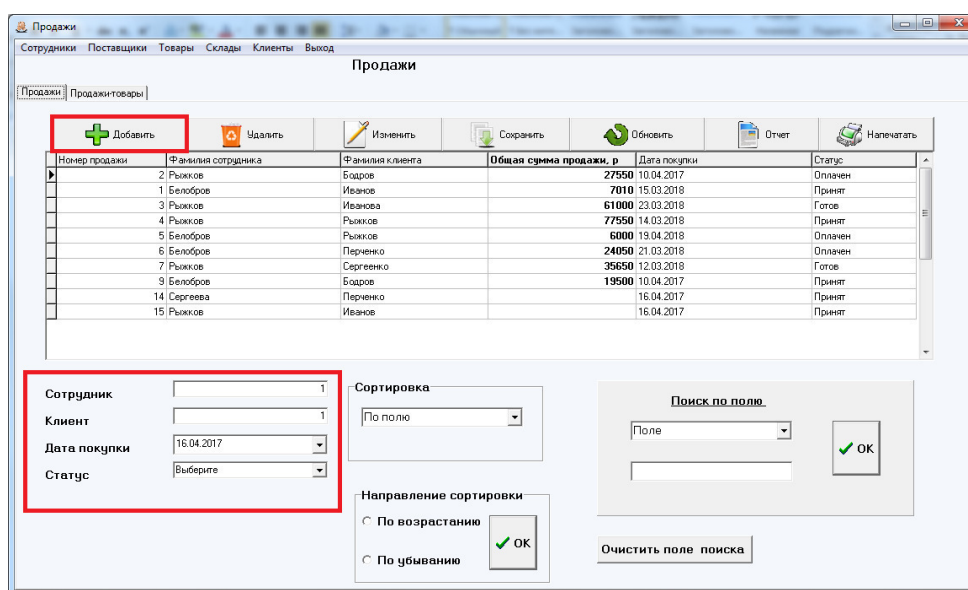


Рисунок 3.37 – Добавление новой покупки

Далее необходимо перейти на вкладку «Продажи-товары» и добавить по данной покупке информацию о купленных товарах и их количествах.

Например, по покупке № 14 клиент приобрел 10 ноутбуков и 2 модема (рисунок 3.38).

После добавления купленных клиентом товаров, необходимо перейти во вкладку «Продажи». По данной продаже появится общая стоимость (рисунок 3.40). После оплаты необходимо изменить статус заказа на «Оплачен».

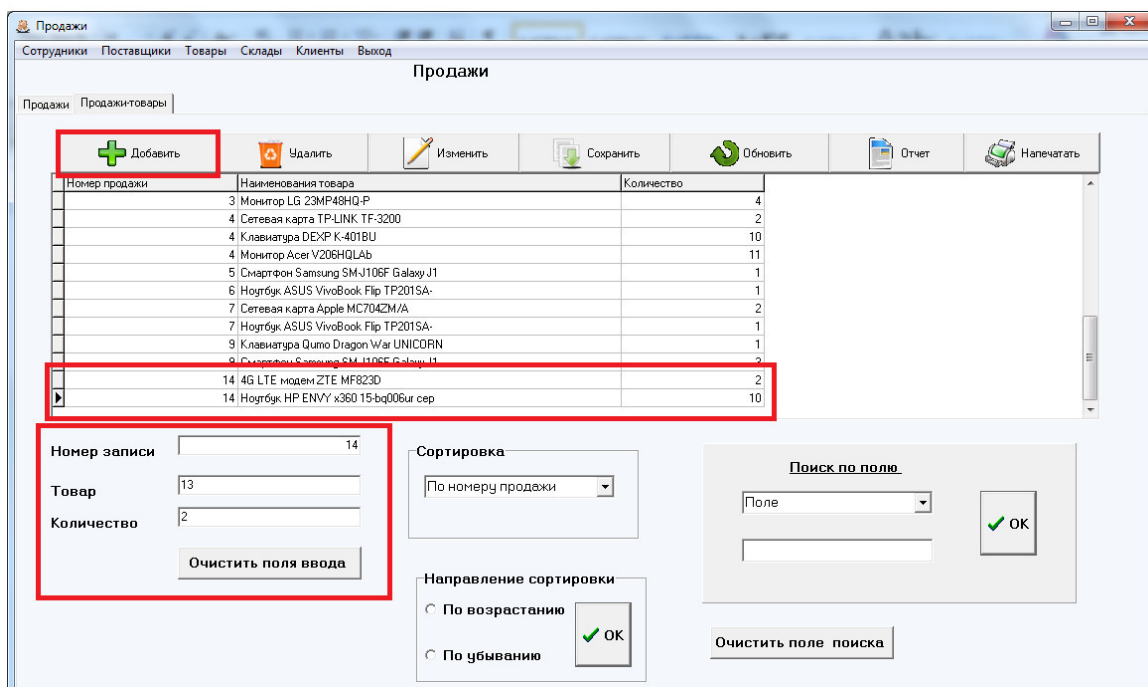


Рисунок 3.38 – Добавление товаров по покупке

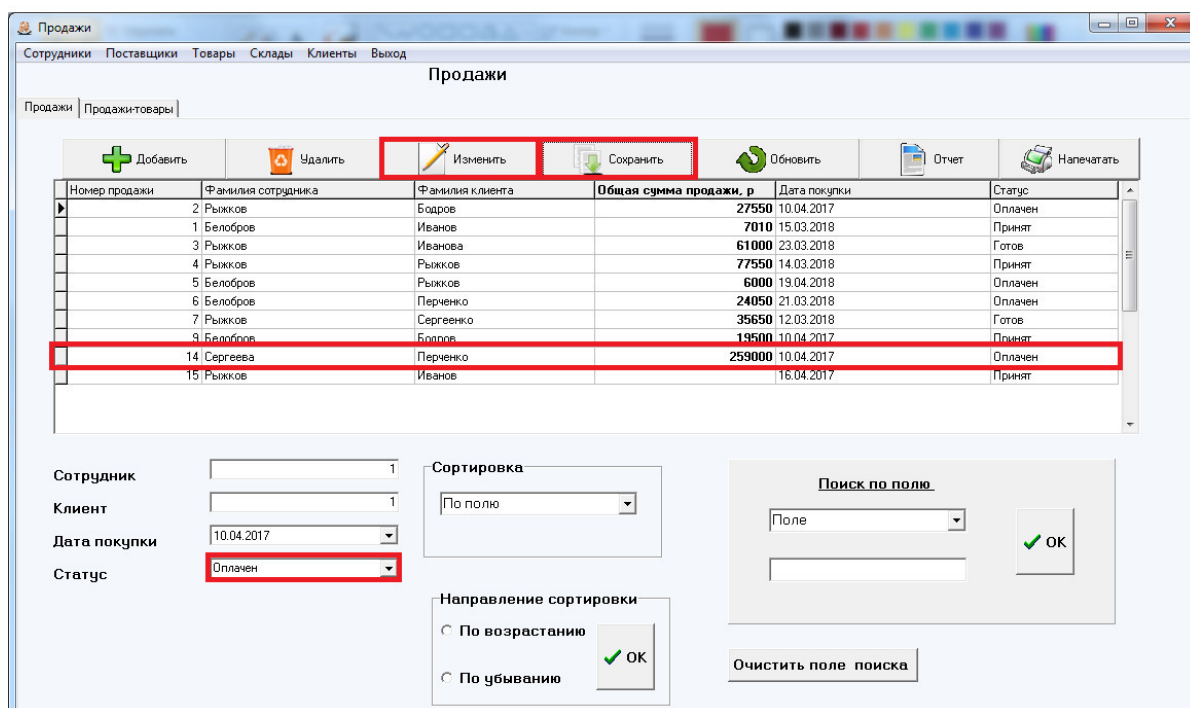


Рисунок 3.39 – Изменение статуса покупки на «Отплачен».

После оплаты заказа необходимо сформировать и распечатать документ продажи для того, чтобы клиент мог предоставить его кладовщику и получить весь купленный товар на складе.

Для этого необходимо перейти во вкладку «Продажи-товары» с помощью кнопки поиска найти все товара по данной покупке (рисунок 3.40). После этого, необходимо нажать кнопку «Отчет». В результате появиться форма, представленная на рисунке 3.41, на которой продавец должен поставить печать.

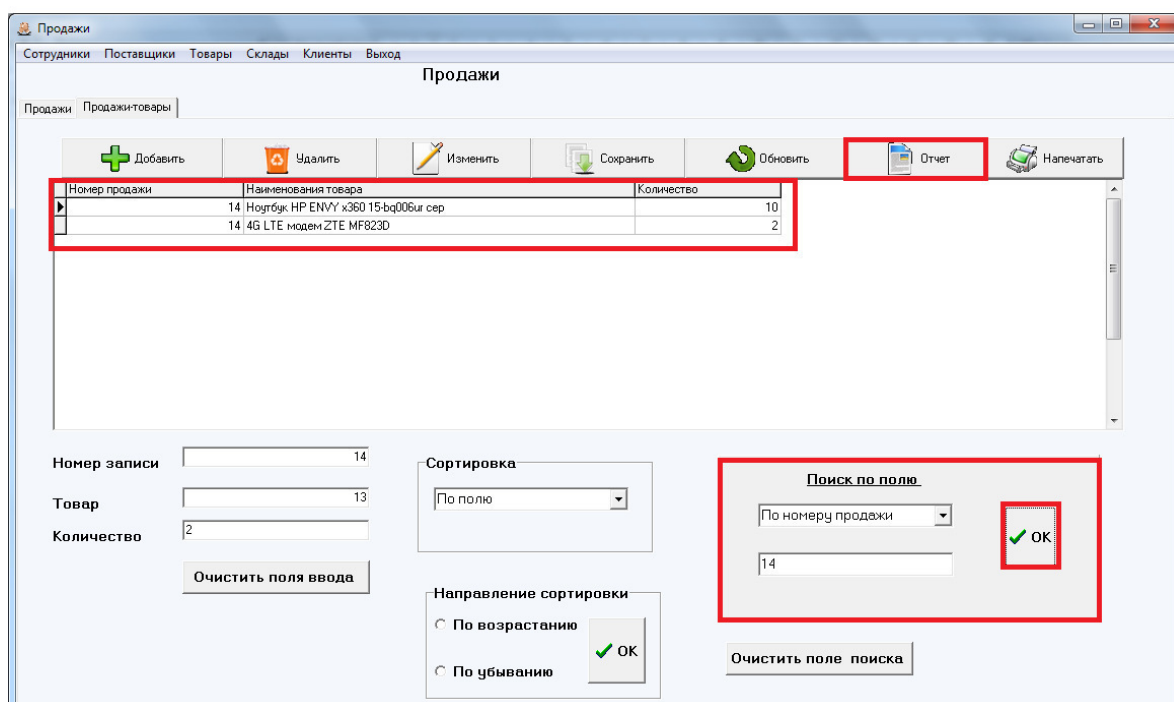


Рисунок 3.40 – Поиск покупки

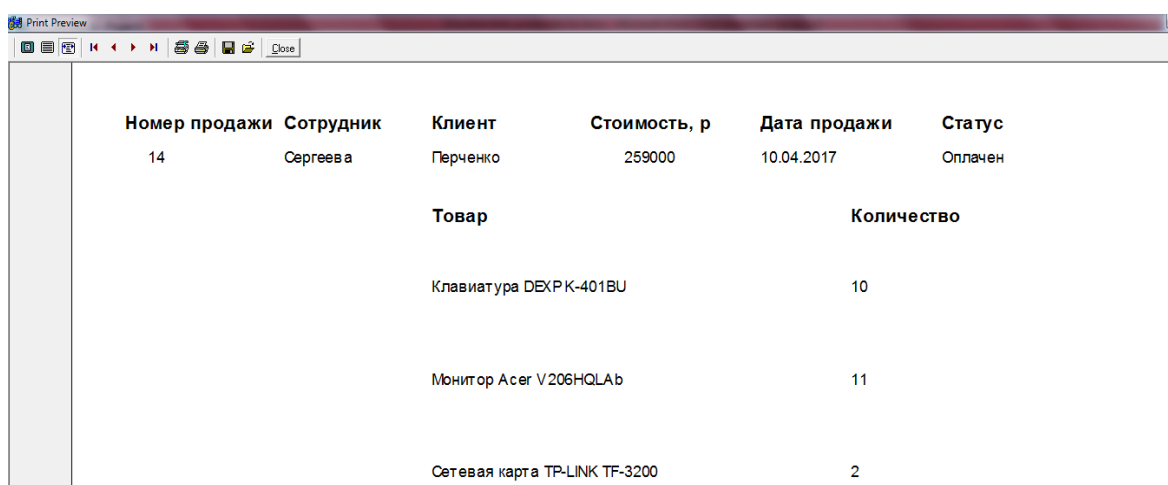


Рисунок 3.41 – Документ продажи

Для анализа предпочтений покупателей система позволяет формировать рейтинг товаров в виде диаграммы. Для её просмотра необходимо перейти в раздел меню «Товары» и нажать кнопку «Просмотреть рейтинг товаров (рисунок 3.42-3.43).

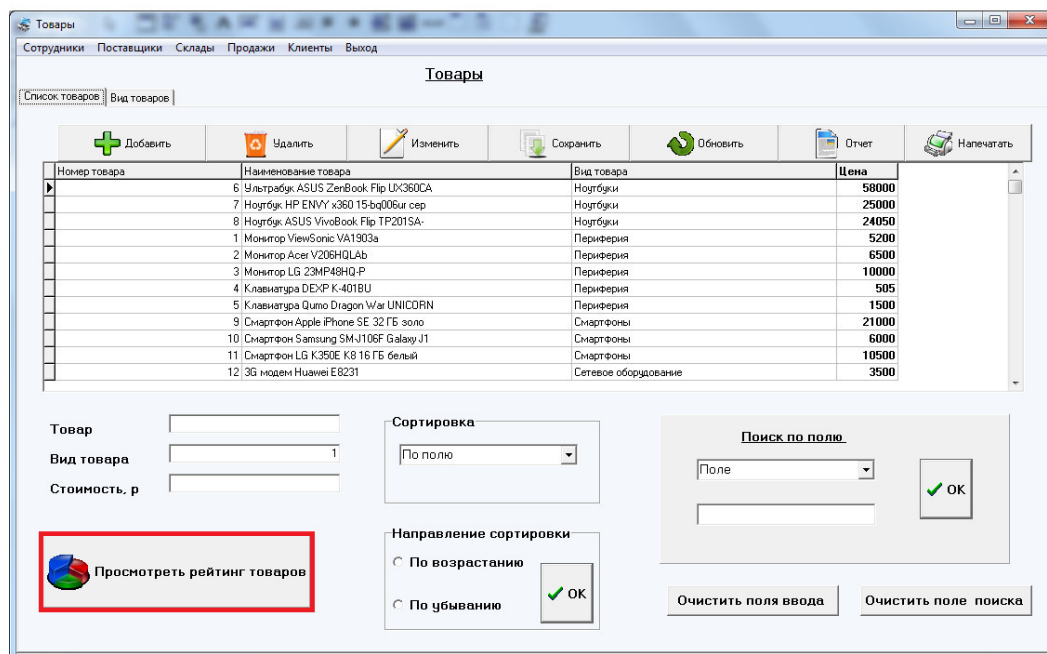


Рисунок 3.42 – Форма «Товары»

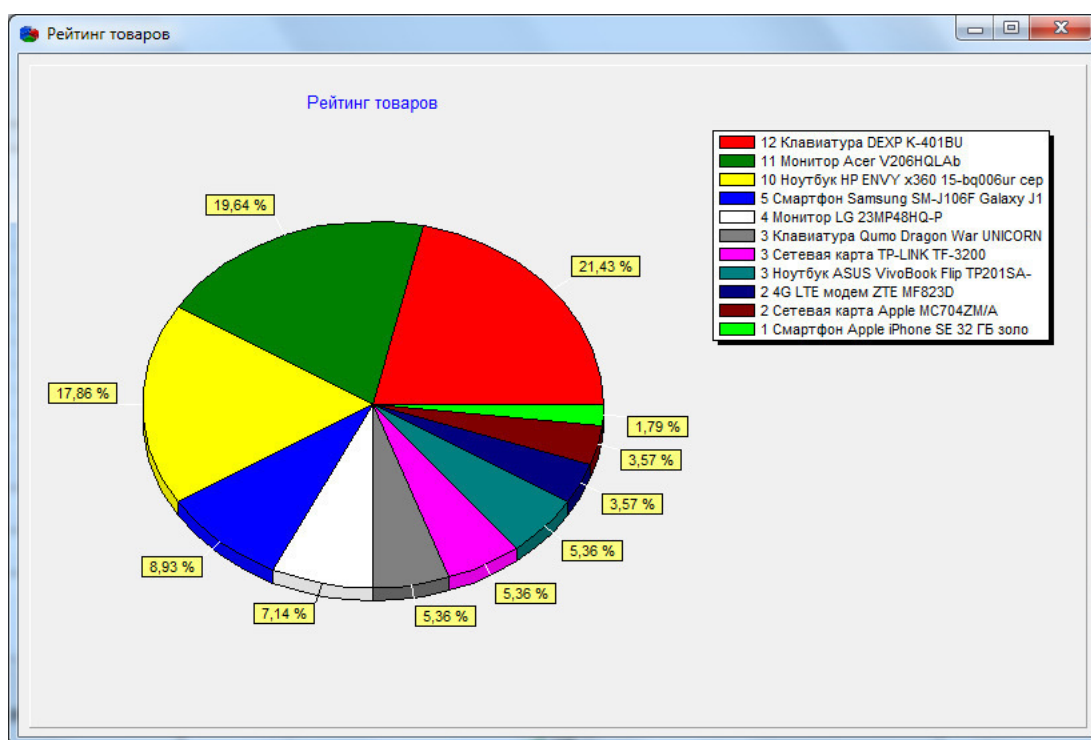


Рисунок 3.43 – Диаграмма рейтинга товаров

3.5 Оценка экономической эффективности разработки

Экономическая эффективность – это соотношение экономического эффекта и затрат, породивших этот эффект [40].

Себестоимость разработки информационной системы определялась по следующему ряду статей:

- основная заработная плата производственного персонала;
- отчисления на социальные нужды;
- затраты на электроэнергию;
- затраты на амортизацию и ремонт вычислительной техники;
- накладные расходы [40].

Прежде всего необходимо произвести расчет затрат на оплату труда разработчиков информационной системы магазина компьютерной техники. Для этого произведем оценку трудоемкости разработки календарным методом. Данные о трудоемкости разработки представлены в таблице 3.1.

Таблица 3.1 – Оценка трудоемкости разработки

№	Наименование этапа	Трудоемкость, часы
1	Исследование деятельности компании «RossComService»	80
2	Проектирование информационной системы	80
3	Выбор методов и средств разработки автоматизированной системы	40
4	Разработка базы данных и пользовательского интерфейса	150
5	Программная реализация автоматизированной системы	170
6	Тестирование и оптимизация информационной системы	80
Итого:		600

Таким образом, трудоемкость разработки автоматизированной системы компьютерного магазина равна 600 часов, то есть 75 рабочим дням при восьмичасовом рабочем дне. Среднемесячный фонд рабочего времени

разработчика 164,16 часов, среднемесячная заработная плата 45000 рублей [40].

Расчет основной заработной платы ($Z_{осн}$) производился по формуле 3.1 [40].

$$Z_{осн} = \frac{Z_{ср}}{\Phi_{ср}} * Ч, \quad (3.1)$$

где $Z_{ср}$ – среднемесячная заработная плата;

$\Phi_{ср}$ – среднемесячный фонд рабочего времени;

$Ч$ – это количество отработанных часов.

В соответствии с формулой 3.1 основная заработная плата разработчика составила 164473,7 рубля.

В соответствии с Федеральным законом от 24 июля 2009 года N 212-ФЗ "О страховых взносах в Пенсионный фонд РФ, Фонд социального страхования РФ, Федеральный фонд обязательного медицинского страхования и территориальные фонды обязательного медицинского страхования" (в редакции Федерального закона от 03.12.2011 № 379-ФЗ) страховой взнос составляет 26% от дохода, который вычисляется по формуле 3.2 [40]:

$$СВ = Z_{осн} * P, \quad (3.2)$$

где $Z_{осн}$ – основная заработная плата;

P – размер страхового взноса на социальные нужды.

Итоговые отчисления на социальные нужды составили 42763,2 рублей, а основная заработная плата разработчика с учетом отчислений – 121710,5 рублей.

Затраты на электроэнергию были рассчитаны по формуле 3.3 [40]:

$$C_{э} = H_q * Ч * T_q, \quad (3.3)$$

где H_q – норма потребления электроэнергии за час;

$Ч$ – количество рабочих часов электроприборов;

T – тарифный план за использование 1 КВт/ч.

Затраты на использование электроэнергии стационарным компьютером и монитором с мощностями 0,07 КВт/ч и 0,04 КВт/ч, тарифным планом – 3,53 рубля за 1 КВт за 600 часов работы составили 233 рубля.

К накладным расходам были отнесены расходы на все виды связи, в том числе на широкополосный доступ к сети Интернет, стоимость которого составляет 450 рублей в месяц за 100 Мбит/с. За 3 месяца разработок общая сумма составит 1350 рублей.

Амортизация – процесс постепенного переноса стоимости средств производства на производимый продукт [40].

Амортизационные отчисления – постепенное возмещение износа основных фондов в стоимостном выражении [40].

Расчет амортизационных отчислений по компьютерной технике вычислялся исходя из нормы амортизации, установленной в зависимости от нормативного срока использования компьютерной техники. Для стационарного компьютера предполагаемый срок использования 5 лет, а процент амортизационных отчислений в год составил 20%. Первоначальная стоимость компьютера устанавливалась исходя из средней стоимости с учетом расходов на доставку, установку и монтаж (35200 рублей), а годовой фонд рабочего времени в часах, исходя из нормы рабочего времени на 2018 год (1970 час).

Расчет амортизационных отчислений вычислялся по формуле 3.4 [40]:

$$AO = \frac{C_n * P_{AO}}{\Phi_z} * \Phi_{cp}, \quad (3.4)$$

где C_n – первоначальная стоимость компьютера;

P_{AO} – процент амортизационных отчислений в год;

Φ_z – годовой фонд рабочего времени в часах за 2018 год;

Φ_{cp} – среднемесячный фонд рабочего времени.

Таким образом, амортизационные отчисления составили 586,6 рублей.

На основе произведенных расчетов по отдельным статьям калькуляции была составлена таблица 3.2.

Таблица 3.2 – Калькуляция себестоимости

№	Наименование статей затрат	Сумма (руб.)
1	Затраты на оплату труда работников	121710,50
2	Отчисления на социальные нужды	42763,20
3	Накладные расходы	1350,00
4	Амортизационные отчисления	606,28
5	Электроэнергия	233,00
	Итого:	166662, 98

В результате себестоимость автоматизированной торговой системы составила 166662, 98 рублей.

Критерием эффективности создания и внедрения информационных систем выступает ожидаемый годовой экономический эффект, получаемый потребителем программы. Чистая прибыль от использования продукта за год определялась по формуле 3.5 [40]:

$$П = P_z - Z_z, \quad (3.5)$$

где P_z – стоимостная оценка результатов применения программного продукта в течение года;

Z_z – стоимостная оценка затрат при использовании программного продукта [40].

При ручном ведении бизнес-процессов магазина компьютерной техники затраты среднее время на проведение одной операции сотрудником составляет 30 минут, а при автоматизированном проведении – всего лишь 3 минуты. То есть за год величина выплат сотруднику при том же объеме работы сократится в 10 раз.

Средняя заработная продавца-консультанта 25000 рублей в месяц, а кладовщика – 30000 рублей, что за год составит 300000 рублей и 360000. После внедрения автоматизированной подсистемы магазина компьютерной техники затраты на оплату труда за тот же объем работы сократятся до 30000 рублей и 42000 рублей за год.

После внедрения системы необходимо ввести новую должность системного администратора, который будет заниматься услугами технической поддержки пользователей системы и администрирование базы данных. Его средняя заработная плата составляет 35000 рублей в месяц, то 420000 рублей в год, 109200 рублей в год – отчисления на социальные нужды и 310800 рублей в год – заработная плата с учетом отчислений.

Отчисления на социальные нужды сотрудников составит 171600 (78000 и 93600) рублей в год, а также 17160 рублей, а чистая заработная плата за год составит 560400 (222000 и 266400) рублей и 56040 рублей за год при ручном и автоматизированном формировании отчета соответственно.

Амортизационные отчисления не изменят своего значения и составят 606,28 рублей в год как до, так и после внедрения автоматизированного отчета.

Накладные расходы также уменьшились в 10 раз, так как объем работы за год остается прежним (доступ к сети Интернет 5400 и 540 рублей за год соответственно).

Затраты на использование электроэнергии стационарным компьютером и монитором с мощностями 0,07 кВт/ч и 0,04 кВт/ч, тарифным планом – 3,53 рублей за 1 кВт, а также фондом рабочего времени 1740 часа за год работы вычислялись по формуле 3.2 составили 675,7 рублей, а после внедрения автоматизированной системы при том же объеме работы – 67,57 рублей.

Сопоставление расходов за год ручного построения отчета и автоматизированного представлено в таблице 3.3.

Таблица 3.3 – Сравнение себестоимости

№ п/п	Наименование статьи	Сумма за год (ручное)	Сумма за год (автоматизированное)
1	Затраты на оплату труда работников	560400,00	366840,00
2	Отчисления на социальные нужды	171600,00	126360,00
3	Накладные расходы	5400,00	540,00
4	Амортизационные отчисления	606,28	606,28
5	Электроэнергия	675,7	67,57
	Итого	738681,98	494413,85

Таким образом, чистая прибыль от использования разработанного программного продукта за год составит 244268,13 рублей.

Для вычисления срока окупаемости использовалась формула 3.6, значение которого составило 0,7 года (менее 9 месяцев) [40].

$$CO = K_0 / C, \quad (3.6)$$

где K_0 – сумма первоначально вложенных средств;

C – годовые поступления от внедрения системы.

Выводы по третьему разделу: разработанная автоматизированная подсистема компьютерного магазина является экономически эффективной, так как прибыль от использования за год – более 200000 рублей, а также чистая прибыль за первый же год покрывает себестоимость разработки. Срок окупаемости разработанной системы составит 9 месяцев.

ЗАКЛЮЧЕНИЕ

В настоящее время большое значение уделяется автоматизации технологических и бизнес-процессов на предприятии. Это напрямую связано с постоянно меняющимся рынком, высокой скоростью в принятии решений, многозадачностью в управлении активами и необходимостью снижения рисков. Руководители предприятий стремятся использовать самые современные подходы к организации хозяйственной деятельности. Выходом в условиях все усложняющейся внутренней и внешней среды предприятия становится комплексная автоматизация бизнес-процессов.

В рамках выпускной квалификационной работы рассматривалась деятельность компьютерного сервиса «RossComService», который в 2017 году расширил область деятельности и теперь занимается также продажей компьютерной техники.

С целью повышения прибыли была разработана автоматизированная подсистема компьютерного магазина. Она обеспечивает взаимодействие сотрудников магазина с клиентами и поставщика, а также позволяет вести складской учет.

Для достижения поставленной цели были решены следующие задачи:

- исследование компании и её бизнес-процессов;
- проектирование структурно-функциональных моделей деятельности «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ»;
- анализ существующих разработок;
- обоснование проектных решений по техническому, информационному, программному и технологическому обеспечению задачи;
- разработка и тестирование программного средства;
- оценка экономической эффективности разработки.

Для построения диаграмм бизнес-процессов и физической и логической моделей базы данных было использовано CASE-средство AllFusion Process Modeler и AllFusion ERwin Data Modeler, которые

позволяют создавать, документировать и сопровождать базы данных, хранилища и витрины данных.

Модели данных помогли визуализировать структуру данных, обеспечили эффективный процесс организации, управления и администрирования таких аспектов деятельности предприятия, как уровень сложности данных, технологий баз данных и среды развертывания.

Одной из важнейших задач являлась разработка пользовательского интерфейса. При разработке использовались стандартные, привычные пользователям элементы, что обеспечивает максимальное удобство. При проектировании интерфейса были получены навыки анализа действий типичного пользователя системы и продуманы исключительные ситуации, возможные во время взаимодействия программного обеспечения и пользователя.

Подсистема компьютерного магазина является экономически эффективной, так как чистая прибыль от использования разработанного программного средства за год превышает затраты на использование системы и за 9 месяцев покрывает её себестоимость.

В будущем планируется интеграция разработанной подсистемы компьютерного магазина с уже функционирующей системой учета сервисных услуг. Интеграция программных систем и продуктов – это обмен данными между системами с возможной последующей их обработкой.

Кроме того, планируется разработка дополнительных аналитических отчетных форм, которые позволят управленческому звену компании более точно прогнозировать развитие торговой деятельности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Демчук, О.Н. Теория организации : Учебное пособие : [Электронный ресурс] / О.Н. Демчук, Т.А. Ефремова. - 3-е изд., стер. - Москва : Флинта, 2017. - 262 с.
2. Асадуллаев, Р.Г. Формально-логические средства организации управления : [Электронный ресурс]: Учебно-методический комплекс / Р.Г. Асадуллаев. - Белгород, 2016.
3. Гахов, Р.П. Компьютерное моделирование экономических процессов [Текст]: учебное пособие для студентов вузов / Р.П. Гахов, Н.В. Щербинина - Белгород: ИД Белгород, 2014. - 88 с.
4. Сибирева, И.Д. Экономика организации (предприятия) : учебное пособие / И.Д. Сибирева; рец.: Т.Н. Флигинских, И.М. Лычева ; РАГС при Президенте РФ, Белгородский фил. ОРАГС. - Белгород : КОНСТАНТА, 2015. - 158 с.
5. Бунеева, Р.И. Коммерческая деятельность: организация и управление: учебник для студентов вузов, обучающихся по спец.: «Экономика и управление на предприятии» / Р.И. Бунеева; рец.: Г.Л. Кругляков, Е.Л. Хрючкина. - Ростов н/Д : Феникс, 2016.
6. Мочалова, Я.В. Организация инновационной деятельности предприятий (организаций): [Электронный ресурс]: Учебно-методический комплекс / Я.В. Мочалова. - Белгород, 2014.
7. Семенихин, В.В. Торговля : Энциклопедия : [Электронный ресурс] / В.В. Семенихин; В.В. Семенихин. - Москва : ГроссМедиа, 2014. - 1140 с.
8. Медведева, О.В. Бухгалтерский учет в торговле : Учебное пособие для студентов вузов / О.В. Медведева, О.Н. Попкова, З.Д. Чернышова; Под ред. О.В. Медведевой; Рец.: А.Л. Черненко, Р.Ю. Симионов, Л.Ф. Капустьянова. - Ростов н/Д : Феникс, 2014. - 606 с.
9. Пугинский, Б.И. Коммерческое право России: учебник для студентов вузов, обучающихся по спец.: 021100 (030501) "Юриспруденция", 030500

(521400) "Юриспруденция (бакалавр)", аспирантов, преподавателей юридических вузов / Б.И. Пугинский. - 4-е изд., перераб. и доп. - М. : Юрайт : Юрайт-Издат, 2010. - 355 с.

10. Учитель, Ю.Г. Разработка управленческих решений: Учебник: [Электронный ресурс] / Ю.Г. Учитель, А.И. Терновой, К.И. Терновой; Ю.Г. Учитель, А.И. Терновой, К.И. Терновой. - 2-е изд., перераб. и доп. - Москва : Юнити-Дана, 2017. - 384 с.

11. Философова, Т.Г. Конкуренция. Инновации. Конкурентоспособность: Учебное пособие: [Электронный ресурс] / Т.Г. Философова, В.А. Быков;; под ред. Т.Г. Философовой. - 2-е изд., перераб. и доп. - Москва : Юнити-Дана, 2014. - 296 с.

12. Маторин, С.И. Теория систем и системный анализ [[Электронный ресурс]] / С.И. Маторин, О.А. Зимовец - Белгород: ИД Белгород, 2016.

13. Ширяев, В.И. Управление бизнес-процессами: Учебно-методическое пособие : [Электронный ресурс] / В.И. Ширяев, Е.В. Ширяев; В.И. Ширяев, Е.В. Ширяев. - Москва : Финансы и статистика, 2016. -258 с.

14. Еналеева, И.Д. Правила торговли. Защита прав потребителя. Защита прав продавца: Справочное пособие. Комментарии / И.Д. Еналеева и др. - М. ; Ростов н/Д : МарТ, 2015. - 448 с.

15. Ефимова, Е.Г. Экономика: [Электронный ресурс]: Учебник / Е.Г. Ефимова. - 4-е изд., стер. - Москва: Флинта, 2018. - 392 с.

16. Ломакин, В.В. Программирование и программное обеспечение информационных технологий [[Электронный ресурс]] / В.В Ломакин – НИУ БелГУ. - Белгород, 2014.

17. Маклаков, С.В. Моделирование бизнес-процессов с BPwin 4.0 [Текст] / С.В. Маклаков. – М.: ДИАЛОГ-МИФИ, 2013. – 224 с.

18. Гахов, Р.П. Методы и средства проектирования информационных систем и технологий: Учебно-методический комплекс [[Электронный ресурс]] / Р.П. Гахов – НИУ БелГУ. - Белгород, 2015.

19. Сирант, О.В. Работа с базами данных : Курс лекций : [Электронный ресурс] / О.В. Сирант, Т.А. Коваленко. - 2-е изд., испр. - Москва: Национальный Открытый Университет "ИНТУИТ", 2016. - 402 с.
20. Гахов, Р.П. Принципы построения, проектирования и эксплуатации автоматизированных информационных систем: [Электронный ресурс]: Учебно-методический комплекс / Р.П. Гахов. - Белгород, 2016.
21. Баженова, И.Ю. Основы проектирования приложений баз данных: Курс лекций: [Электронный ресурс] / И.Ю. Баженова. - 2-е изд., испр. - Москва: Национальный Открытый Университет "ИНТУИТ", 2016. - 238 с.
22. Антонов, В.Ф. Методы и средства проектирования информационных систем:[Электронный ресурс]: Учебное пособие / В.Ф. Антонов, А.А. Москвитин. - Ставрополь : СКФУ, 2016. - 342 с.
23. Кузнецов, С. Ю. Введение в реляционные базы данных : Курс лекций : [Электронный ресурс] / С.Ю. Кузнецов. - 2-е изд., испр. - Москва : Национальный Открытый Университет "ИНТУИТ", 2016. - 248 с.
24. Баженова, И.Ю. SQL и процедурно-ориентированные языки : Курс лекций : [Электронный ресурс] / И.Ю. Баженова. - 2-е изд., испр. - Москва : Национальный Открытый Университет "ИНТУИТ", 2016. - 167 с.
25. Манаенкова Т.А. Управление базами о данных клиентах : [Электронный ресурс] : Учебно-методический комплекс / Т.А. Манаенкова, В.В. Ломакин. - Белгород, 2015.
26. Колокольникова, А.И. Компьютерное моделирование финансовой деятельности: Учебное пособие: [[Электронный ресурс]] / А.И. Колокольникова - Москва: Директ-Медиа, 2016. - 164 с.
27. Репин, В.И. Бизнес-процессы. Моделирование, внедрение, управление [[Электронный ресурс]] / В.И. Репин - Москва: Флинта, 2017. - 480 с.
28. Золотов, С.Ю. Проектирование информационных систем [[Электронный ресурс]]: учебное пособие / С. Ю. Золотов - Томск: Эль Контент, 2016. - 88 с.

29. Гахова, Н.Н. Инструментальные средства информационных систем [[Электронный ресурс]] / Н.Н. Гахова - Белгород: НИУ БелГУ, 2015. - Режим доступа: <http://pegas.bsu.edu.ru/course/view.php?id=5188>
30. Сеницын, С.В. Верификация программного обеспечения [[Электронный ресурс]]: курс лекций / С. В. Сеницын, Н. Ю. Налютин - Москва: Интернет-Университет Информационных Технологий, 2017. - 367 с.
31. Золотов, С.Ю. Проектирование информационных систем [[Электронный ресурс]] / С.Ю Золотов – Томск: Эль Контент, 2015. - 88 с.
32. Кубенский, А.А. Функциональное программирование : [Электронный ресурс]: Учебник и практикум для академического бакалавриата / А.А. Кубенский. - Москва : Юрайт, 2018. - 348 с.
33. Огнева, М.В. Программирование на языке C++ : практический курс : Учебное пособие для бакалавриата и специалитета : [Электронный ресурс] / М.В. Огнева, Е.В. Кудрина. - Москва : Юрайт, 2018. - 335 с.
34. Фридман, А.Л. Основы объектно-ориентированного программирования на языке Си++ / А.Л. Фридман. - 3-е изд. - М. : Горячая линия - Телеком, 2017. - 234 с.
35. Шамраев, А.А. Математические методы в инженерных расчетах: Учебно-методический комплекс : [Электронный ресурс] / А.А. Шамраев, Е.О. Шамраева. - Белгород, 2017.
36. Туманов, В.Е. Проектирование хранилищ данных для систем бизнес-аналитики : Учебное пособие : [Электронный ресурс] / В.Е. Туманов; В.Е. Туманов. - Москва : Интернет-Университет Информационных Технологий, 2015. - 616 с.
37. Гудов, А.М. Базы данных и системы управления базами данных. Программирование на языке PL/SQL : Учебное пособие: [Электронный ресурс] / А.М. Гудов, С.Ю. Завозкин, Т.С. Рейн; А.М. Гудов, С.Ю. Завозкин, Т.С. Рейн. - Кемерово : КемГУ, 2015. - 134 с.

38. Енин, А.В. Локальная СУБД своими руками : Учимся на примерах: [Электронный ресурс] / А.В. Енин, Н.В. Енин; А.В. Енин, Н.В. Енин. – Москва: СОЛОН-ПРЕСС, 2016. - 464 с.

39. Шалыгина, Н.П. Экономика организации (предприятия): [Электронный ресурс]: Учебно-методический комплекс / Н.П. Шалыгина. - Белгород, 2014.

40. Карышев, М.Ю. Социально-экономическая эффективность сферы информационно-коммуникационных технологий: методология международных статистических сопоставлений : Монография : [Электронный ресурс] / М.Ю. Карышев; М.Ю. Карышев. - Москва : Финансы и статистика, 2014. - 160 с.

41. Бородин, Р.А. Особенности учета затрат на производство и калькуляции себестоимости продукции [[Электронный ресурс]] / Р.А. Бородин. - Москва: Лаборатория книги, 2014. - 141 с.

ПРИЛОЖЕНИЕ А

Функциональные диаграммы «КАК ЕСТЬ»

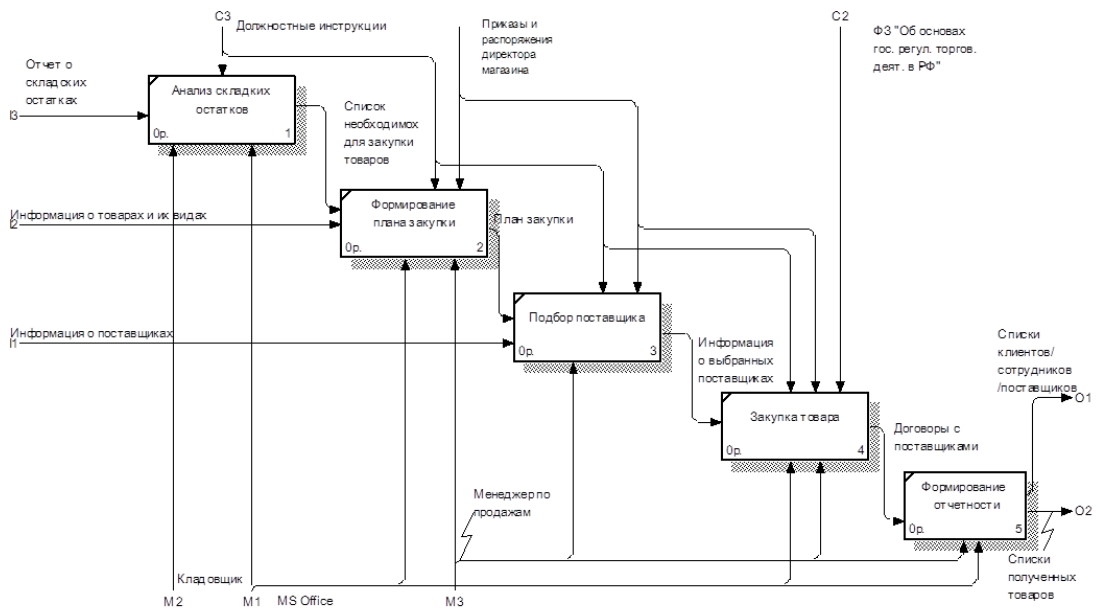


Рисунок А.1 – Диаграмма-декомпозиция «Закупка товаров»

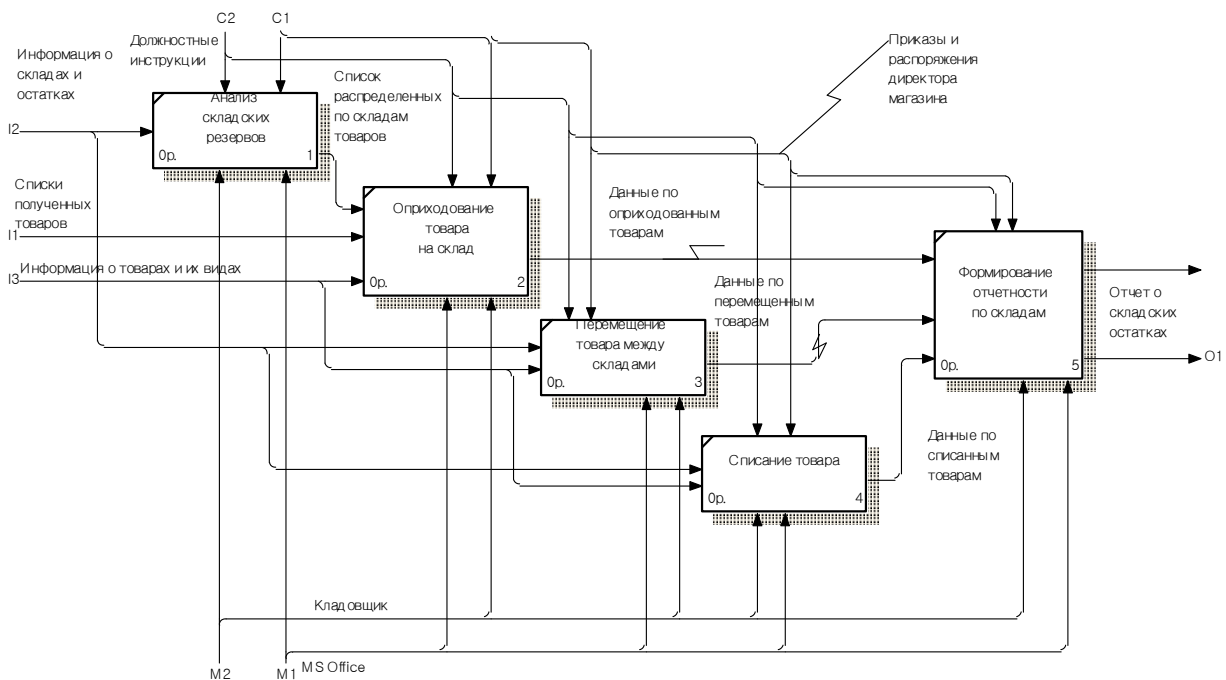


Рисунок А.2 – Диаграмма-декомпозиция «Складской учет»

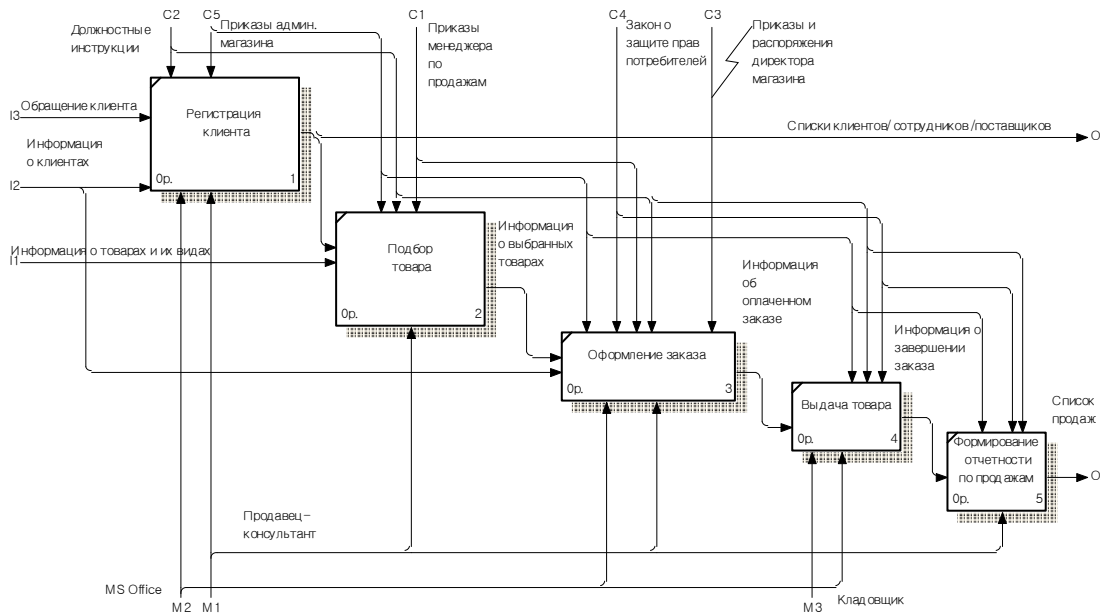


Рисунок А.3 – Диаграмма-декомпозиция «Продажа компьютерной техники»

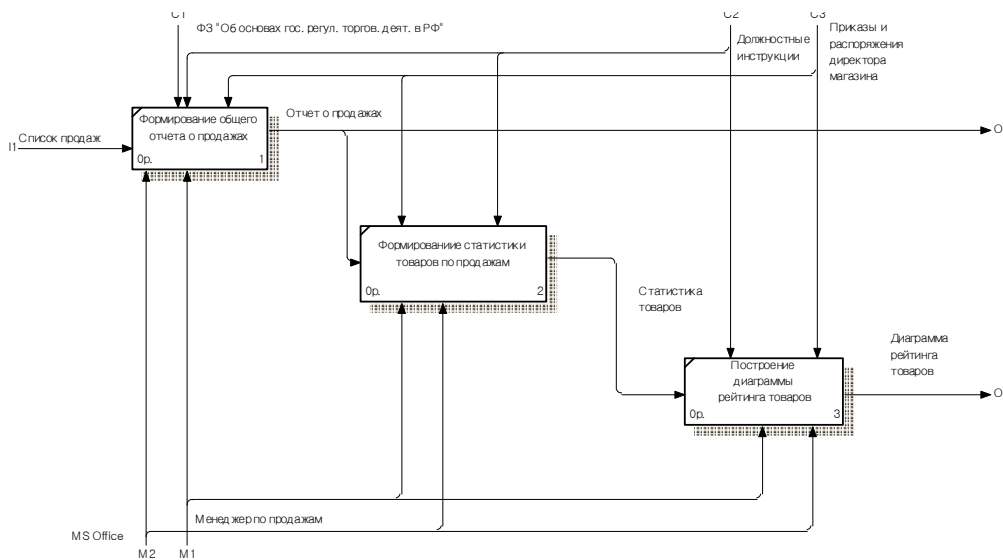


Рисунок А.4 – Диаграмма-декомпозиция «Анализ торговой деятельности магазина»

ПРИЛОЖЕНИЕ Б

Функциональные диаграммы «КАК ДОЛЖНО БЫТЬ»

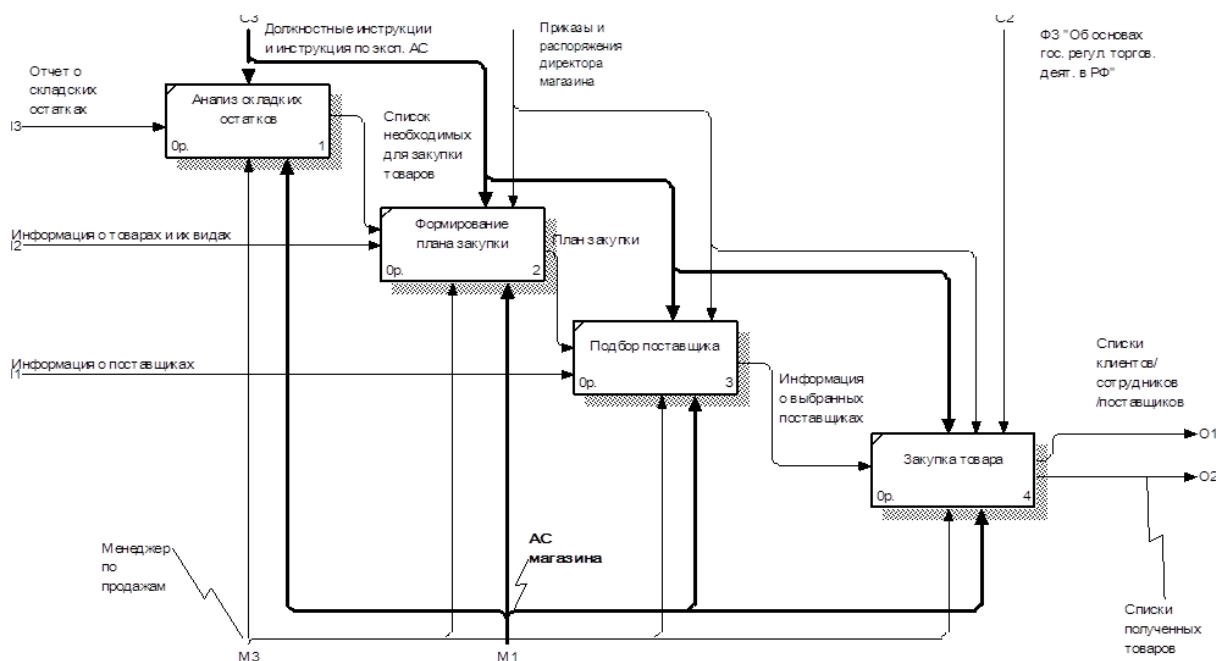


Рисунок Б.1 - Диаграмма-декомпозиция «Закупка товара»

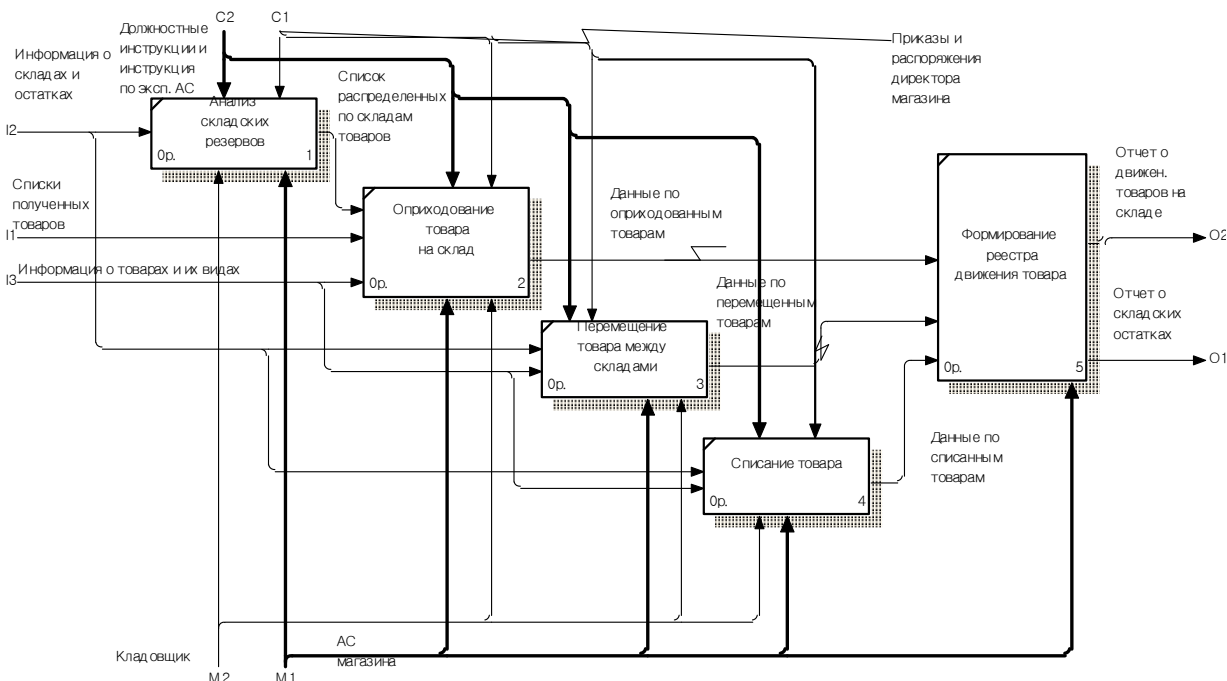


Рисунок А.2 – Диаграмма-декомпозиция «Складской учет»

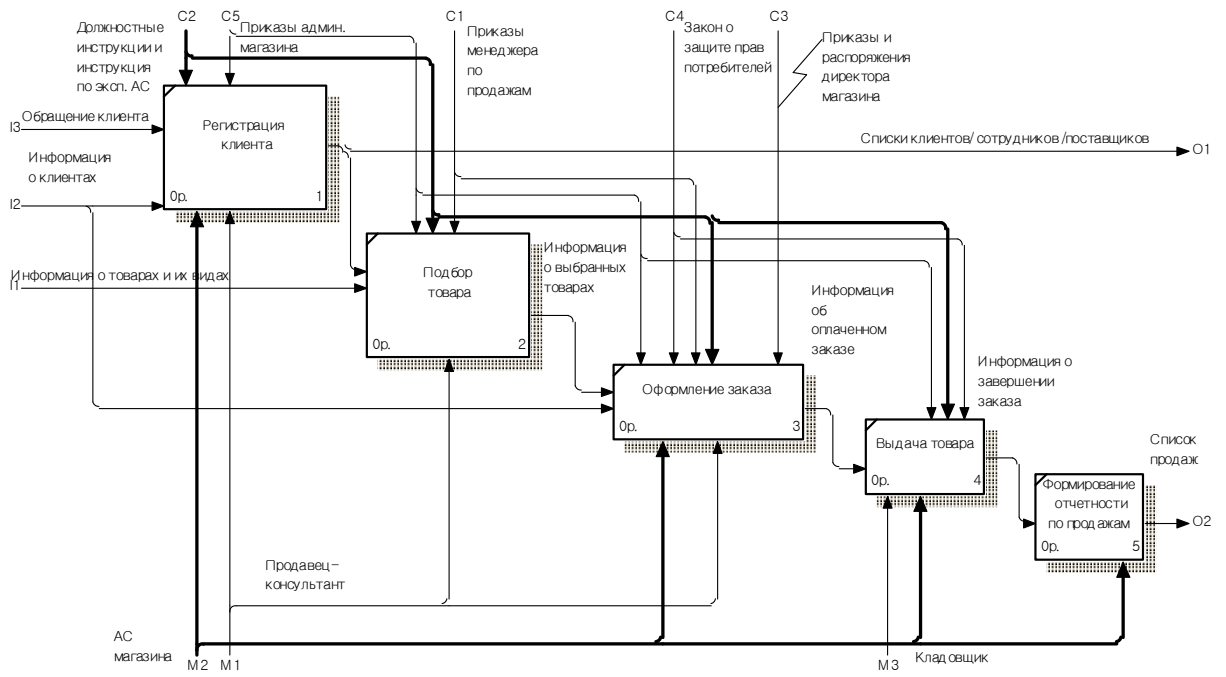


Рисунок Б.3 – Диаграмма-декомпозиция «Продажа компьютерной техники»

ПРИЛОЖЕНИЕ В

Фрагмент sql-кода базы данных

```
SET SQL DIALECT 3;
SET NAMES WIN1251;

CREATE DATABASE 'localhost:D:\Диплом
\SHOP_COMP.FDB'
USER 'SYSDBA' PASSWORD 'masterkey'
PAGE_SIZE 16384
DEFAULT CHARACTER SET WIN1251;
/****
****/
/*****
*****/
CREATE DOMAIN D_CAB AS
CHAR(8);
CREATE DOMAIN D_DATE AS
DATE;

CREATE DOMAIN D_ID AS
INTEGER
NOT NULL;

CREATE DOMAIN D_INSUR AS
CHAR(16)
NOT NULL;

CREATE DOMAIN D_NAME AS
CHAR(35);

CREATE DOMAIN D_PHONE AS
CHAR(11);

CREATE DOMAIN D_PRICE AS
FLOAT
NOT NULL;

CREATE DOMAIN D_SMALL AS
SMALLINT;

CREATE DOMAIN D_SNILS AS
CHAR(11)
NOT NULL;

CREATE DOMAIN D_TIME AS
TIME;

CREATE DOMAIN STATUS AS
CHAR(10);

/****
*****/
/****
****/
Domains
CREATE GENERATOR GEN_CL;
SET GENERATOR GEN_CL TO 17;

CREATE GENERATOR GEN_EMP;
SET GENERATOR GEN_EMP TO 8;

CREATE GENERATOR GEN_MOV;
SET GENERATOR GEN_MOV TO 0;

CREATE GENERATOR GEN_PAT;
SET GENERATOR GEN_PAT TO 12;

CREATE GENERATOR GEN_PEOP;
SET GENERATOR GEN_PEOP TO 36;

CREATE GENERATOR GEN_POST;
SET GENERATOR GEN_POST TO 6;

CREATE GENERATOR GEN_PROD;
SET GENERATOR GEN_PROD TO 21;

CREATE GENERATOR GEN_PROV;
SET GENERATOR GEN_PROV TO 12;

CREATE GENERATOR GEN_REC;
SET GENERATOR GEN_REC TO 10;

CREATE GENERATOR GEN_REG;
SET GENERATOR GEN_REG TO 16;

CREATE GENERATOR GEN_SELL;
SET GENERATOR GEN_SELL TO 16;

CREATE GENERATOR GEN_SER;
SET GENERATOR GEN_SER TO 34;

CREATE GENERATOR GEN_STOCK;
SET GENERATOR GEN_STOCK TO 4;

CREATE GENERATOR GEN_TICK;
SET GENERATOR GEN_TICK TO 33;

CREATE GENERATOR GEN_TYPES;
SET GENERATOR GEN_TYPES TO 8;

CREATE GENERATOR GEN_TYPE_PROD;
SET GENERATOR GEN_TYPE_PROD TO 16;

/****
*****/
/****
****/
Generators
SET TERM ^ ;
```

```

/*****
****
/****
****
/****
****
****
****

ID_EMPLOYEES D_ID,
COUNT_PR D_ID,
DATA_REG D_DATE,
INDEX_R D_ID)
AS
BEGIN
EXIT;
END^

CREATE PROCEDURE ADD_EMPLOYEERS (
  ID_POST D_ID)
AS
BEGIN
EXIT;
END^

CREATE PROCEDURE ADD_PEOPLE (
  SURNAME D_NAME,
  NAME D_NAME,
  PATRONUMIC D_NAME,
  PHONE D_PHONE,
  ADRESS D_NAME,
  DATE_OF_BIRTH D_DATE,
  PASSPORT D_NAME)
AS
BEGIN
EXIT;
END^

CREATE PROCEDURE ADD_POST (
  NAME D_NAME)
AS
BEGIN
EXIT;
END^

CREATE PROCEDURE ADD_PROD (
  NAME_PROD D_NAME,
  ID_TYPE D_ID,
  PRICE D_PRICE)
AS
BEGIN
EXIT;
END^

CREATE PROCEDURE ADD_PROV (
  NAME_ORG D_NAME,
  ADRESS D_NAME,
  TELEPHONE D_PHONE,
  CHECK_ACC CHAR(20))
AS
BEGIN
EXIT;
END^

CREATE PROCEDURE ADD_REG (
  ID_MOVE D_ID,
  ID_PRODUCT D_ID,
  ID_STOCK D_ID,

ID_EMPLOYEES D_ID,
COUNT_PR D_ID,
DATA_REG D_DATE,
INDEX_R D_ID)
AS
BEGIN
EXIT;
END^

CREATE PROCEDURE ADD_SELL (
  ID_EMPLOYEES D_ID,
  ID_CLIENT D_ID,
  DATE_SELL D_DATE,
  STATYS_SELL D_NAME)
AS
BEGIN
EXIT;
END^

CREATE PROCEDURE ADD_SELL_PROD (
  ID_SELLING D_ID,
  ID_PRODUCT D_ID,
  COUNT_PROD D_ID)
AS
BEGIN
EXIT;
END^

CREATE PROCEDURE ADD_STOCK (
  STOCK D_NAME)
AS
BEGIN
EXIT;
END^

CREATE PROCEDURE ADD_TYP_PROD (
  "TYPE" D_NAME)
AS
BEGIN
EXIT;
END^

CREATE PROCEDURE ADD_TYPES (
  TYPES D_NAME)
AS
BEGIN
EXIT;
END^

CREATE PROCEDURE DEL_CL (
  ID_CLIENT D_ID)
AS
BEGIN
EXIT;
END^

```



```

CREATE PROCEDURE DEL_EMP (
  ID_EMPLOYEES D_ID)
AS
BEGIN
  EXIT;
END^

```

```

CREATE PROCEDURE DEL_PAT (
  ID_PAT D_ID)
AS
BEGIN
  EXIT;
END^

```

```

CREATE PROCEDURE DEL_POST (
  ID_POST D_ID)
AS
BEGIN
  EXIT;
END^

```

```

CREATE PROCEDURE DEL_PROD (
  ID_PRODUCT D_ID)
AS
BEGIN
  EXIT;
END^

```

```

CREATE PROCEDURE DEL_PROV (
  ID_PROVIDER D_ID)
AS
BEGIN
  EXIT;
END^

```

```

CREATE PROCEDURE DEL_REC (
  ID_REC D_ID)
AS
BEGIN
  EXIT;
END^

```

```

CREATE PROCEDURE DEL_SELL (
  ID_SELLING D_ID)
AS
BEGIN
  EXIT;
END^

```

```

CREATE PROCEDURE DEL_SELL_PROD (
  ID_SELLING D_ID,
  NAME_PROD D_NAME)
AS
BEGIN

```

```

  EXIT;
END^

```

```

CREATE PROCEDURE DEL_STOCK (
  ID_STOCK D_ID)
AS
BEGIN
  EXIT;
END^

```

```

CREATE PROCEDURE DEL_TYPE (
  ID_TYPES D_ID)
AS
BEGIN
  EXIT;
END^

```

```

CREATE PROCEDURE DEL_TYPE_PROD (
  ID_TYPE D_ID)
AS
BEGIN
  EXIT;
END^

```

```

CREATE PROCEDURE UP_EMPLOYEES (
  ID_EMPLOYEES D_ID,
  POST D_NAME)
AS
BEGIN
  EXIT;
END^

```

```

CREATE PROCEDURE UP_PEOPLE (
  ID_PEOPLE D_ID,
  SURNAME D_NAME,
  NAME D_NAME,
  PATRONUMIC D_NAME,
  PHONE D_PHONE,
  ADRESS D_NAME,
  DATE_OF_BIRTH D_DATE,
  PASSPORT D_NAME)
AS
BEGIN
  EXIT;
END^

```

```

CREATE PROCEDURE UP_POST (
  ID_POST D_ID,
  POST D_NAME)
AS
BEGIN
  EXIT;
END^

```

ПРИЛОЖЕНИЕ Г

Фрагмент кода пользовательского приложения

```
#include <vcl.h>
#pragma hdrstop
#include "File1.h"
#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
#include "Unit4.h"
#include "Unit5.h"
#include "Unit6.h"
#include "Unit7.h"
#include "Unit8.h"
#include "Unit9.h"
#include "Unit10.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
int flag;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----

void __fastcall TForm1::BitBtn1Click(TObject
*Sender)
{
Form2->Show();
Form2->TabSheet1->Show();
if ((flag == 1) || (flag == 5)){
Form2->N2->Visible=true;
Form2->N3->Visible=true;
Form2->N4->Visible=true;
Form2->N5->Visible=true;
Form2->N6->Visible=true;
Form2->N7->Visible=true;
Form2->N8->Visible=true;
Form2->N9->Visible=true;
Form2->N10->Visible=true;
//Form2->N11->Visible=true;
Form2->N12->Visible=true;
Form2->N13->Visible=true;
}
Form1->Visible = false;
}

//-----
void __fastcall TForm1::FormClose(TObject *Sender,
TCloseAction &Action)
{
Form3->Close();
}
//-----

void __fastcall TForm1::N1Click(TObject *Sender)
{
Form1->Visible = false;
Form3->Show();
}
//-----

void __fastcall TForm1::BitBtn2Click(TObject
*Sender)
{
Form4->Show();
if ((flag == 1) || (flag == 5)){
Form4->N3->Visible=true;
Form4->N4->Visible=true;
Form4->N5->Visible=true;
Form4->N6->Visible=true;
Form4->N7->Visible=true;
Form4->N8->Visible=true;
Form4->N9->Visible=true;
Form4->N10->Visible=true;
//Form4->N11->Visible=true;
Form4->N12->Visible=true;
Form4->N13->Visible=true;
} else if ((flag == 2) || (flag == 4)){
Form4->N1->Visible=false;
Form4->N12->Visible=false;
}
Form1->Visible = false;
}
//-----

void __fastcall TForm1::BitBtn3Click(TObject
*Sender)
{
if ((flag == 2) || (flag == 4)){
Form5->N1->Visible=false;
Form5->N12->Visible=false;
}
Form5->Show();
}
//-----
```

```

Form5->TabSheet2->Show();
Form1->Visible = false;
}
//-----
-----

void __fastcall TForm1::BitBtn4Click(TObject
*Sender)
{
if ((flag == 2) || (flag == 4)){
Form5->N1->Visible=false;
Form5->N12->Visible=false;
}
Form5->Show();
Form5->TabSheet1->Show();
Form1->Visible = false;
}
//-----
-----

void __fastcall TForm1::BitBtn5Click(TObject
*Sender)
{
if ((flag == 2) || (flag == 4)){
Form6->N1->Visible=false;
Form6->N12->Visible=false;
}
Form6->Show();
Form6->TabSheet1->Show();
Form1->Visible = false;
}
//-----
-----

void __fastcall TForm1::BitBtn6Click(TObject
*Sender)
{
if ((flag == 2) || (flag == 4)){
Form6->N1->Visible=false;
Form6->N12->Visible=false;
}
Form6->Show();
//Form7->TabSheet1->Show();

Form1->Visible = false;
}
//-----
-----

void __fastcall TForm1::BitBtn7Click(TObject
*Sender)
{
if ((flag == 2) || (flag == 4)){
Form7->N1->Visible=false;
Form7->N12->Visible=false;
}
Form7->Show();
Form7->TabSheet2->Show();
Form1->Visible = false;
}
//-----
-----

void __fastcall TForm1::BitBtn8Click(TObject
*Sender)
{
if ((flag == 2) || (flag == 4)){
Form7->N1->Visible=false;
} else if ( flag == 3){
Form7->N1->Visible=false;
Form7->N2->Visible=false;
Form7->N3->Visible=false;
Form7->N4->Visible=false;
Form7->N5->Visible=false;
Form7->N6->Visible=false;
Form7->N7->Visible=false;
Form7->N8->Visible=false;
//Form7->N9->Visible=false;
//Form7->N10->Visible=false;
//Form8->N11->Visible=false;
Form7->N13->Visible=true;
}
Form7->Show();
Form1->Visible = false;
}
//-----
-----
#include <vel.h>
#pragma hdrstop

#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
#include "Unit4.h"
#include "Unit5.h"
#include "Unit6.h"
#include "Unit7.h"
#include "Unit8.h"
#include "Unit9.h"
#include "Unit10.h"
#include "Unit12.h"
#include "File1.h"
//-----
-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm2 *Form2;
//-----
-----
__fastcall TForm2::TForm2(TComponent* Owner)
: TForm(Owner)
{
}
//-----
-----

void __fastcall TForm2::FormClose(TObject *Sender,
TCloseAction &Action)
{
Form3->Close();
}
//-----
-----

void __fastcall TForm2::N13Click(TObject *Sender)
{

```

```

Form2->Visible = false;
Form3->Show();
}
//-----
-----

void __fastcall TForm2::FormCreate(TObject *Sender)
{
//DataModule9->DataSource1->DataSet->Delete();
switch (flag)
{
    case 2:
        Form2->N13->Visible = false;
        break;
    //case 3:
    //case 4:
    //case 5:
}
DateTimePicker1->Date=Now().DateString();
}
//-----
-----

void __fastcall TForm2::BitBtn1Click(TObject
*Sender)
{

if ((Form2->Edit1->Text != "" ) && (Form2->Edit2-
>Text != "" ) && (Form2->Edit3->Text != "" ) &&
(Form2->Edit4->Text != "" ) && (Form2->Edit16-
>Text != "" ) && (Form2->Edit6->Text != "" )){
DataModule9->IBStoredProc1->ParamByName
("SURNAME")->AsString=Edit1->Text;
DataModule9->IBStoredProc1->ParamByName
("NAME")->AsString=Edit2->Text;
DataModule9->IBStoredProc1->ParamByName
("PATRONUMIC")->AsString=Edit3->Text;
DataModule9->IBStoredProc1->ParamByName
("PHONE")->AsString=Edit16->Text;
DataModule9->IBStoredProc1->ParamByName
("ADRESS")->AsString=Edit4->Text;
DataModule9->IBStoredProc1->ParamByName
("DATE_OF_BIRTH")->AsString=DateTimePicker1-
>Date;
DataModule9->IBStoredProc1->ParamByName
("PASSPORT")->AsString=Edit6->Text;
DataModule9->IBStoredProc1->Prepare();
DataModule9->IBStoredProc1->ExecProc();
DataModule9->IBStoredProc2->ParamByName
("ID_POST")->AsString=DBEdit1->Text;
DataModule9->IBStoredProc2->Prepare();
DataModule9->IBStoredProc2->ExecProc();
DataModule9->IBQuery1->Close();
DataModule9->IBQuery1->Open();
}else{
    ShowMessage("Заполните все поля!");
}

}
//-----
-----

void __fastcall TForm2::N2Click(TObject *Sender)
{
Form4->Show();
Form2->Visible = false;
}
//-----
-----

void __fastcall TForm2::N4Click(TObject *Sender)
{
Form2->Visible = false;
Form5->Show();
Form5->TabSheet2->Show();
}
//-----
-----

void __fastcall TForm2::N5Click(TObject *Sender)
{
Form2->Visible = false;
Form5->Show();
Form5->TabSheet1->Show();
}
//-----
-----

void __fastcall TForm2::N7Click(TObject *Sender)
{
Form2->Visible = false;
Form6->Show();
Form6->TabSheet1->Show();
}
//-----
-----

void __fastcall TForm2::N8Click(TObject *Sender)
{
Form2->Visible = false;
Form6->Show();
Form6->TabSheet2->Show();
}
//-----
-----

void __fastcall TForm2::N12Click(TObject *Sender)
{
Form2->Visible = false;
Form8->Show();
}
//-----
-----

void __fastcall TForm2::DBEdit1Click(TObject
*Sender)
{

Form10->Show();
Form10->TabSheet2->Show();
}

```

```

}
//-----
void __fastcall TForm2::N16Click(TObject *Sender)
{
Form2->Visible = false;
Form7->Show();
//Form7->TabSheet1->Show();
}
//-----

void __fastcall TForm2::N10Click(TObject *Sender)
{
Form2->Visible = false;
Form7->Show();
Form7->TabSheet2->Show();
}
//-----

void __fastcall TForm2::N11Click(TObject *Sender)
{
Form2->Visible = false;
Form7->Show();
Form7->TabSheet3->Show();
}
//-----

void __fastcall TForm2::ComboBox1Click(TObject
*Sender)
{
Form2->ComboBox2->Text="Поле";
Edit7->Clear();
}
//-----

void __fastcall TForm2::ComboBox2Click(TObject
*Sender)
{
Form2->ComboBox1->Text="По полю";
}
//-----

void __fastcall TForm2::Button2Click(TObject
*Sender)
{
Edit1->Clear();
Edit2->Clear();
Edit3->Clear();
Edit16->Clear();
Edit4->Clear();
Edit6->Clear();
}
//-----

void __fastcall TForm2::Button1Click(TObject
*Sender)
{
Edit7->Clear();
}
//-----

void __fastcall TForm2::ComboBox3Click(TObject
*Sender)
{
Form2->ComboBox4->Text="Поле";
Edit14->Clear();
}
//-----

void __fastcall TForm2::ComboBox4Click(TObject
*Sender)
{
Form2->ComboBox3->Text="По полю";
}
//-----

void __fastcall TForm2::Button3Click(TObject
*Sender)
{
Edit8->Clear();
}
//-----

void __fastcall TForm2::Button4Click(TObject
*Sender)
{
Edit14->Clear();
}
//-----

void __fastcall TForm2::BitBtn8Click(TObject
*Sender)
{
AnsiString d;
switch (RadioGroup1->ItemIndex)
{case 0: d = " "; break;
case 1: d = " DESC";
}
DataModule9->IBQuery1->Close();
DataModule9->IBQuery1->SQL->Clear();
DataModule9->IBQuery1->SQL->Add("select * from
view_emp");

switch (ComboBox1->ItemIndex)
{case 0: d= "ORDER BY ID_EMPLOYEES"+ d; break;
case 1: d= "ORDER BY SURNAME"+ d; break;
case 2: d= "ORDER BY NAME"+ d; break;
case 3: d= "ORDER BY PATRONUMIC"+ d; break;
case 5: d= "ORDER BY DATE_OF_BIRTH"+ d;
break;
case 4: d= "ORDER BY POST_NAME"+ d; break;
}
}
}

```

```

DataModule9->IBQuery1->SQL->Add(d);
DataModule9->IBQuery1->Open();
}
//-----
-----

void __fastcall TForm2::BitBtn9Click(TObject
*Sender)
{
DataModule9->IBQuery1->Close();
DataModule9->IBQuery1->SQL->Clear();

switch (ComboBox2->ItemIndex)
{
case 0:
DataModule9->IBQuery1->SQL->Add("select *
from view_emp where ID_EMPLOYEES like '%" +
Edit7->Text + "%'");
break;
case 1:
DataModule9->IBQuery1->SQL->Add("select *
from view_emp where SURNAME like '%" + Edit7-
>Text + "%'");
break;
case 2:
DataModule9->IBQuery1->SQL->Add("select *
from view_emp where NAME like '%" + Edit7->Text +
"%'");
break;
case 3:
DataModule9->IBQuery1->SQL->Add("select *
from view_emp where PATRONOMIC like '%" +
Edit7->Text + "%'");
break;
case 5:
DataModule9->IBQuery1->SQL->Add("select *
from view_emp where DATE_OF_BIRTH like '%" +
Edit7->Text + "%'");
break;
case 4:
DataModule9->IBQuery1->SQL->Add("select *
from view_emp where POST_NAME like '%" + Edit7-
>Text + "%'");
break;
}
DataModule9->IBQuery1->Open();
if ((DataModule9->IBQuery1->RecordCount==0) ||
(Form2->Edit7->Text == ""))
{
ShowMessage("Не найдено!");
}
}
//-----
-----

void __fastcall TForm2::BitBtn5Click(TObject
*Sender)
{
DataModule9->IBQuery1->Close();
DataModule9->IBQuery1->SQL->Clear();
DataModule9->IBQuery1->SQL->Add("select * from
view_emp");

```

```

DataModule9->IBQuery1->Open();
Form2->ComboBox2->Text="Поле";
Edit7->Clear();
Form2->ComboBox1->Text="По полю";
Label5->Font->Color=clWindowText;
DBEdit1->Font->Color=clWindowText;
}
//-----
-----

void __fastcall TForm2::BitBtn17Click(TObject
*Sender)
{
AnsiString d;
switch (RadioGroup2->ItemIndex)
{case 0: d = " "; break;
case 1: d = " DESC";
}
DataModule9->IBQuery2->Close();
DataModule9->IBQuery2->SQL->Clear();
DataModule9->IBQuery2->SQL->Add("select * from
post");

switch (ComboBox3->ItemIndex)
{case 0: d= "ORDER BY ID_post"+ d; break;
case 1: d= "ORDER BY name"+ d; break;
}
DataModule9->IBQuery2->SQL->Add(d);
DataModule9->IBQuery2->Open();
}
//-----
-----

void __fastcall TForm2::BitBtn18Click(TObject
*Sender)
{
DataModule9->IBQuery2->Close();
DataModule9->IBQuery2->SQL->Clear();

switch (ComboBox4->ItemIndex)
{
case 0:
DataModule9->IBQuery2->SQL->Add("select *
from post where ID_post like '%" + Edit14->Text + "%'
");
break;
case 1:
DataModule9->IBQuery2->SQL->Add("select *
from post where NAME like '%" + Edit14->Text + "%'
");
break;
}
DataModule9->IBQuery2->Open();
if ((DataModule9->IBQuery2->RecordCount==0) ||
(Form2->Edit14->Text == ""))
{
ShowMessage("Не найдено!");
}
}
//-----
-----

```

```

void __fastcall TForm2::BitBtn14Click(TObject
*Sender)
{
DataModule9->IBQuery2->Close();
DataModule9->IBQuery2->SQL->Clear();
DataModule9->IBQuery2->SQL->Add("select * from
post");
DataModule9->IBQuery2->Open();
Form2->ComboBox4->Text="Поле";
Edit14->Clear();
Form2->ComboBox3->Text="По полю";
}
//-----

void __fastcall TForm2::BitBtn10Click(TObject
*Sender)
{
if (Form2->Edit8->Text != "" ) {
DataModule9->IBStoredProc3->ParamByName
("NAME")->AsString=Edit8->Text;
DataModule9->IBStoredProc3->Prepare();
DataModule9->IBStoredProc3->ExecProc();
DataModule9->IBQuery2->Close();
DataModule9->IBQuery2->Open();
} else {
ShowMessage("Заполните все поля!");
}
}
//-----

void __fastcall TForm2::BitBtn2Click(TObject
*Sender)
{
DataModule9->IBStoredProc12->ParamByName
("id_employees")->AsString=DBEdit2->Text;
DataModule9->IBStoredProc12->Prepare();
DataModule9->IBStoredProc12->ExecProc();
DataModule9->IBQuery1->Close();
DataModule9->IBQuery1->Open();
}
//-----

void __fastcall TForm2::BitBtn11Click(TObject
*Sender)
{
DataModule9->IBStoredProc13->ParamByName
("id_post")->AsString=DBEdit3->Text;
DataModule9->IBStoredProc13->Prepare();
DataModule9->IBStoredProc13->ExecProc();
DataModule9->IBQuery2->Close();
DataModule9->IBQuery2->Open();
}
//-----

void __fastcall TForm2::BitBtn3Click(TObject
*Sender)
{

```

```

ShowMessage("Внимание! Вы перешли в режим
редактирования! Заполните обязательные поля,
выделенные КРАСНЫМ цветом. Для отмены
нажмите кнопку ОБНОВИТЬ. ");
Label5->Font->Color=clRed;
DBEdit1->Font->Color=clRed;
}
//-----

void __fastcall TForm2::BitBtn4Click(TObject
*Sender)
{
DataModule9->IBStoredProc22->ParamByName
("id_people")->AsString=DBEdit2->Text;
if (Edit1->Text != ""){
DataModule9->IBStoredProc22-
>ParamByName ("SURNAME")->AsString=Edit1-
>Text;
} else {
DataModule9->IBStoredProc22-
>ParamByName ("SURNAME")->AsString=DBEdit4-
>Text;
}
if (Edit2->Text != ""){
DataModule9->IBStoredProc22-
>ParamByName ("NAME")->AsString=Edit2->Text;
} else {
DataModule9->IBStoredProc22-
>ParamByName ("NAME")->AsString=DBEdit5-
>Text;
}
if (Edit3->Text != ""){
DataModule9->IBStoredProc22-
>ParamByName ("PATRONUMIC")->AsString=Edit3-
>Text;
} else {
DataModule9->IBStoredProc22-
>ParamByName ("PATRONUMIC")-
>AsString=DBEdit6->Text;
}
if (Edit16->Text != ""){
DataModule9->IBStoredProc22-
>ParamByName ("PHONE")->AsString=Edit16->Text;
} else {
DataModule9->IBStoredProc22-
>ParamByName ("PHONE")->AsString=DBEdit8-
>Text;
}
if (Edit4->Text != ""){
DataModule9->IBStoredProc22-
>ParamByName ("ADRESS")->AsString=Edit4->Text;
} else {
DataModule9->IBStoredProc22-
>ParamByName ("ADRESS")->AsString=DBEdit9-
>Text;
}
if (DateTimePicker1->Date
!=
Now().DateString()){
DataModule9->IBStoredProc22-
>ParamByName ("DATE_OF_BIRTH")-
>AsString=DateTimePicker1->Date;
} else {

```

```

        DataModule9->IBStoredProc22-
>ParamByName      ("DATE_OF_BIRTH")-
>AsString=DBEdit10->Text;
    }
    if (Edit6->Text!=""){
        DataModule9->IBStoredProc22-
>ParamByName      ("PASSPORT")->AsString=Edit6-
>Text;
    }else{
        DataModule9->IBStoredProc22-
>ParamByName      ("PASSPORT")-
>AsString=DBEdit11->Text;
    }
    DataModule9->IBStoredProc22->Prepare();
    DataModule9->IBStoredProc22->ExecProc();
    DataModule9->IBStoredProc23->ParamByName
("id_employees")->AsString=DBEdit2->Text;

    DataModule9->IBStoredProc23->ParamByName
("post")->AsString=DBEdit1->Text;
    DataModule9->IBStoredProc23->Prepare();
    DataModule9->IBStoredProc23->ExecProc();
    DataModule9->IBQuery1->Close();
    DataModule9->IBQuery1->Open();
    DateTimePicker1->Date = Now();
    Label5->Font->Color=clWindowText;
    DBEdit1->Font->Color=clWindowText;
}
//-----
-----

void __fastcall TForm2::BitBtn12Click(TObject
*Sender)
{
    ShowMessage("Внимание! Вы перешли в режим
редактирования! Для отмены нажмите кнопку
ОБНОВИТЬ.");
}
//-----
-----

void __fastcall TForm2::BitBtn13Click(TObject
*Sender)
{
    DataModule9->IBStoredProc26->ParamByName
("id_post")->AsString=DBEdit3->Text;
    if (Edit8->Text != ""){
        DataModule9->IBStoredProc26-
>ParamByName ("post")->AsString=Edit8->Text;
    }else{
        DataModule9->IBStoredProc26-
>ParamByName ("post")->AsString=DBEdit12->Text;
    }

    DataModule9->IBStoredProc26->Prepare();
    DataModule9->IBStoredProc26->ExecProc();
    DataModule9->IBQuery2->Close();
    DataModule9->IBQuery2->Open();
    DataModule9->IBQuery1->Close();
    DataModule9->IBQuery1->Open();
}
//-----
-----

void __fastcall TForm2::BitBtn6Click(TObject
*Sender)
{
    Form12->QuickRep1->Preview();
}
//-----
-----

void __fastcall TForm2::BitBtn7Click(TObject
*Sender)
{
    Form12->QuickRep1->Print();
}
//-----
-----

void __fastcall TForm2::BitBtn15Click(TObject
*Sender)
{
    Form12->QuickRep3->Preview();
}
//-----
-----

void __fastcall TForm2::BitBtn16Click(TObject
*Sender)
{
    Form12->QuickRep3->Print();
}
//-----
-----

```


Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

«___» _____ г.
