

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки 38.03.05 – Бизнес-информатика
Направленность (профиль) образовательной программы: Электронный бизнес

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
« ____ » _____ 201_ г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Проектирование автоматизированной информационной системы ИТ-отдела ООО «Эрлант»

Исполнитель студент группы 356 об	_____	Г.А. Ковальский
	(подпись, дата)	
Руководитель доцент	_____	И.М. Акилова
	(подпись, дата)	
Консультант по экономической части доцент, канд. техн. наук	_____	Л.В. Рыбакова
	(подпись, дата)	
Нормоконтроль инженер кафедры	_____	В.В. Романико
	(подпись, дата)	

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ
Зав.кафедрой
_____ А.В. Бушманов
« ____ » _____ 2017 г.

З А Д А Н И Е

К бакалаврской работе студента Ковальский Георгий Александрович

1. Тема бакалаврской работы: Проектирование автоматизированной информационной системы ИТ-отдела ООО «Эрлант»

(утверждена приказом от _____ № _____)

2. Срок сдачи студентом законченной работы _____

3. Исходные данные к бакалаврской работе: отчет по практике, специальная литература, нормативные документы.

4. Содержание бакалаврской работы (перечень подлежащих разработке вопросов): анализ объекта исследования, анализ организационной структуры, анализ бизнес-процессов, анализ документооборота, проектирование базы данных, техническое задание, расчет экономической эффективности.

5. Перечень материалов приложения: (наличие чертежей, таблиц, графиков, схем, программных продуктов, иллюстративного материала и т.п.) техническое задание, организационная структура, диаграммы DFD и IDEF0, ER-диаграммы, экранные формы.

6. Консультанты по бакалаврской работе (с указанием относящихся к ним разделов)
консультант по экономической части доцент, канд. техн. наук Рыбакова Л.В.

7. Дата выдачи задания _____

Руководитель бакалаврской работы: доцент, Акилова И.М.

Задание принял к исполнению: _____

РЕФЕРАТ

Бакалаврская работа содержит 66 с., 47 рисунков, 23 таблицы, 20 источников, 2 приложения.

БАЗА ДАННЫХ, СИСТЕМА УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ, УЧЕТ, КОМПЬЮТЕРНАЯ, ТЕХНИКА, ОБСЛУЖИВАНИЕ, РАЗРАБОТКА, ПРОЕКТ, ОТЧЕТЫ

Объектом исследования является система учета комплектующих и обслуживания компьютерной техники и разработки проектов на предприятии.

Цель работы – разработать автоматизированную информационную систему учета комплектующих и обслуживания компьютерной техники и разработки проектов на предприятии.

В процессе работы проведен анализ деятельности ООО «Эрлант», изучены принципы ведения учета разработки проектов и комплектующих, разработана функциональная модель системы, проведено инфологическое проектирование, разработана структура базы данных. В процессе работы использованы CASE- средства BPWin и ERWin.

Разработка автоматизированной системы осуществлялась в среде Delphi 7.0 с использованием СУБД FIREBIRD SQL.

В результате проведенной работы разработана автоматизированная система, позволяющей заносить и обрабатывать сведения о комплектующих компьютеров и копировальной техники и сервисных работах, закупаемых и расходуемых комплектующих в базу данных, вести учет разработки проектов, выполнять необходимые расчетные операции и составлять отчетные документы.

Эффективность разработки заключается в сокращении времени обработки первичных документов, формирования отчетности и повышении производительности труда операторов системы.

СОДЕРЖАНИЕ

Введение	8
1 Анализ деятельности предприятия	9
1.1 Общая характеристика ООО «Эрлант»	9
1.2 Организационная структура предприятия	9
1.3 Функциональная структура предприятия	11
1.4 Анализ основных экономических показателей предприятия	13
1.5 Документооборот предприятия	14
1.5.1 Внешний документооборот	15
1.5.2 Внутренний документооборот	17
2 Проектирование информационной системы	19
2.1 Назначение и цели создания системы	19
2.2 Выбор среды разработки	20
2.3 Функциональная структура системы	22
2.4 Модули системы	23
2.5 Проектирование базы данных	25
2.5.1 Инфологическое проектирование	30
2.5.2 Логическое проектирование	38
2.5.3 Физическое проектирование	40
2.6 Описание структуры программы	45
2.7 Описание интерфейса	46
2.8 Входные и выходные данные	56
3 Обоснование экономической эффективности проекта	57
3.1 Прямая и косвенная экономическая эффективность	57
3.2 Определение затрат на разработку и внедрение программного продукта	59
3.2.1 Стоимость материалов	59
3.2.2 Основная заработная плата	59
3.2.3 Накладные расходы	60

3.3 Выбор метода расчета	61
3.3.1 Расчет текущих затрат	61
Заключение	64
Библиографический список	66
Приложение А Техническое задание	67
Приложение Б Фрагменты исходного кода программы	72

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей бакалаврской работе использованы ссылки на следующие стандарты и нормативные документы:

ГОСТ 19.101-77 ЕСПД Виды программ и программных документов

ГОСТ 19.201-78 ЕСПД Техническое задание, требования к содержанию и оформлению

ГОСТ 19.401-78 ЕСПД Текст программы. Требования к содержанию и оформлению

ГОСТ 19.402-78 ЕСПД Описание программы

ГОСТ 19.505-79 ЕСПД Руководство оператора. Требования к содержанию и оформлению

ГОСТ 19.701-90 ЕСПД Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

АС – автоматизированная система;

БД – база данных;

ЛВС – локальная вычислительная сеть;

НФ – нормальная форма;

ООО – общество с ограниченной ответственностью;

ООП – объектно-ориентированное программирование;

ОС – операционная система;

ПО – программное обеспечение;

РФ – Российская Федерация;

СУБД – система управления базами данных;

УПФР – Управление пенсионным фондом России;

УФНС – Управление Федеральной налоговой службы.

ВВЕДЕНИЕ

Данный проект направлен на автоматизацию учета и обслуживания компьютеров по рабочим местам предприятия, а также учета внутрифирменной разработки проектов, которые реализуются сотрудниками предприятия.

Одним из факторов, определяющих уровень развития современного общества и его интеллектуальные возможности, является оснащённость его средствами вычислительной техники. Сфера использования ЭВМ в настоящее время настолько широка, что нет такой области, где применение ЭВМ было бы нецелесообразным.

Рациональное и умелое использование возможностей ЭВМ является одной из серьёзных проблем настоящего периода развития общества, и актуальность решения этой проблемы растёт по мере увеличения парка ЭВМ и совершенствования их технического и программного оснащения. Эффективный путь решения указанной проблемы состоит в широком использовании на практике автоматизированных информационных систем.

Проблема ведения качественного и своевременного учета была актуальной всегда. Один из способов повышения качества учета – его автоматизация при помощи средств вычислительной техники. Целью автоматизации учета документов в организации является создание условий для наиболее эффективной работы сотрудников с применением вычислительной техники для сбора и автоматической обработки информации.

Целью выпускной квалификационной работы является проектирование автоматизированной информационно-управляющей системы учета компьютерной техники и разрабатываемых проектов в ИТ отделе, анализ существующей системы учета документов и потоков данных, возникающих между сотрудниками предприятия при осуществлении ремонта и обслуживания компьютерной техники, а также определение экономической целесообразности автоматизации учета в организации.

1 АНАЛИЗ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ

1.1 Общая характеристика ООО «Эрлант»

ООО «Эрлант» является дистрибьютором крупнейших компаний по производству сетевого и телекоммуникационного оборудования, таких как «Cisco», «HP», «TP-Link» и «ZyXEL».

В ООО «Эрлант» основным направлением деятельности компании является продажа, установка и сервисное обслуживание сетевого и телекоммуникационного оборудования.

Целью ООО «Эрлант» является продажа и сервисное обслуживание оборудования, которые делают жизнь пользователей каждого потребителя более энергичной и насыщенной, предлагая ему более высокую скорость, мобильность и потенциал.

1.2 Организационная структура предприятия

Организационная структура управления представляет собой совокупность специализированных функциональных подразделений, взаимосвязанных в процессе обоснования, выработки, принятия и реализации управленческих решений.

В составе ООО «Эрлант» выделяется шесть отделов. Организационная структура предприятия представлена на рисунке 1:

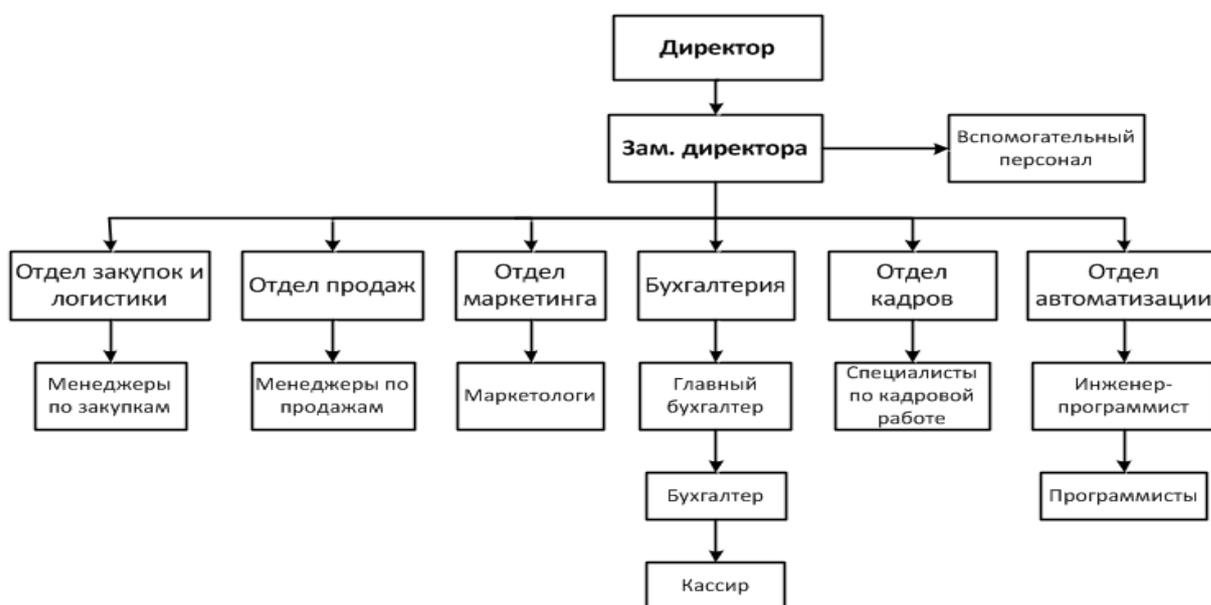


Рисунок 1 – Организационная структура ООО «Эрлант»

Организацию возглавляет директор, который организует всю работу организации и несет полную ответственность за её состояние и деятельность, а также за создание необходимых условий работы сотрудников. В ООО «Эрлант» также работает бухгалтерия, отдел закупок и логистики, отдел продаж, отдел маркетинга финансовый, ИТ отдел и отдел кадров.

Заместитель директора контролирует работу сотрудников, организует их для выполнения текущих задач, руководит процессами перемещения оборудования, следит за его безопасностью торгового оборудования. Кроме того, заместитель директора принимает претензии клиентов, если таковые возникают в отношении качества товара или качества работы сотрудников. В отсутствие директора именно заместитель исполняет его обязанности.

Отдел закупок и логистики занимается заказом продаваемого оборудования, обслуживания, а также решением всех сопутствующих задач.

Отдел продаж занимается поиском и обслуживанием клиентов, заключением договоров, также менеджеры по продажам берут на себя обязанности по общению с клиентами по поводу обслуживания оборудования.

Отдел маркетинга выполняет следующие функции:

- проведение анализа рынка (потребителей и конкурентов);
- разработка стратегии деятельности организации на рынке;
- осуществление рекламной деятельности организации;
- планирование поставок с учетом ситуации на рынке;
- расширение рынка потребителей продукции предприятия.

Бухгалтерия занимается всеми финансовыми потоками организации.

Отдел кадров занимается планированием потребности предприятия в персонале, привлечением, отбором и оценкой персонала. Также данный отдел планирует затраты на персонал, на социальные услуги, предоставляемые персоналу, разрабатывает и внедряет системы заработной платы, определяет особенности оплаты труда отдельных категорий работников, занятых на предприятии.

Основными направлениями деятельности ИТ-отдела:

- обслуживание и ремонт компьютерной, оргтехники;

- разработка и сопровождение программного обеспечения;
- техническая поддержка проводимых мероприятий;

Вспомогательный персонал имеет следующие должностные обязанности:

- выполнение любых вспомогательных и подсобных работ на вверенных участках;
- выполнение разгрузки и загрузки спецтехники;
- содержание в чистоте и при необходимости проведение уборки полов, стен, окон вверенных помещений;
- наблюдение за работоспособностью освещения вверенных участков.

1.3 Функциональная структура предприятия

Функциональная структура определяется как совокупность устойчивых операций и процедур, а также их связей, ориентированных на конечный результат, жизненно важный с точки зрения всей организации или ее отдельных частей.

На контекстной диаграмме, представленной на рисунке 2, отображаются внешние объекты и информационные потоки, определяющие деятельность предприятия.

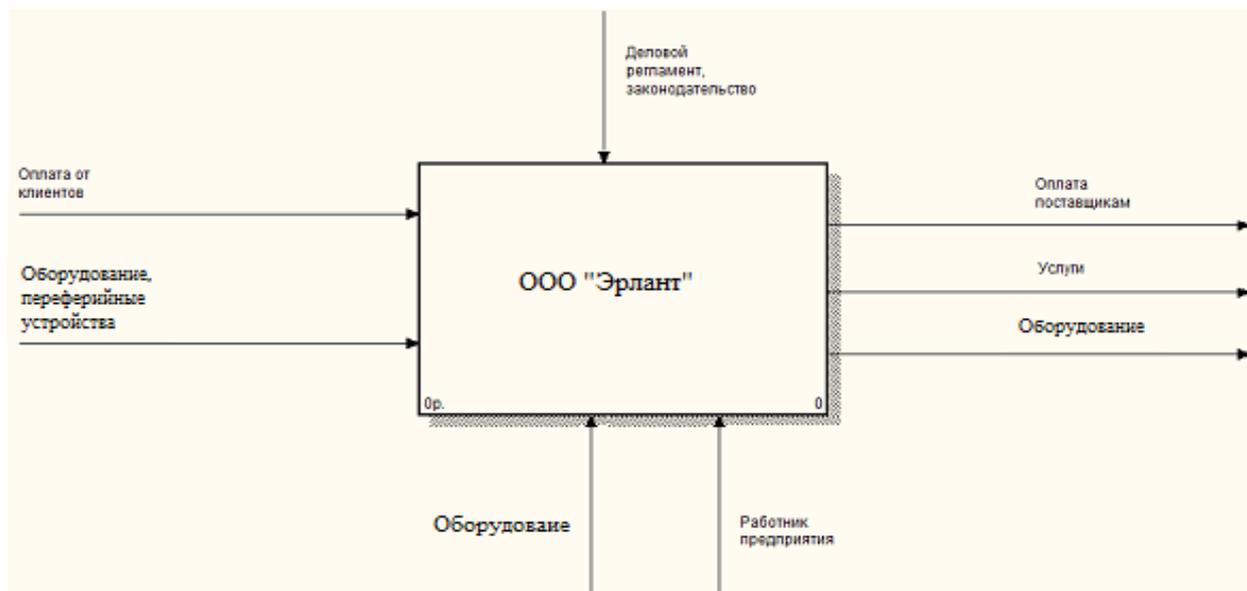


Рисунок 2 – Контекстная диаграмма деятельности ООО «Эрлант»

Основными входящими потоками для компании являются оплата от клиентов, а также оборудование, периферийные устройства и товар. К основным

выходящим потокам относятся: оплата поставщикам, выполненные услуги и проданный товар.

На контекстной диаграмме также отражены управления и механизмы. В роли управления выступает деловой регламент и законодательство РФ. В роли механизмов выступают сотрудники.

Функционирование рассматриваемого предприятия можно разделить на функционирование следующих бизнес-процессов: обработка закупок, бухгалтерский учёт, сервисное обслуживание и обработка продаж.

На данный момент учет компьютеров на предприятии в основном ведется в электронных таблицах EXCEL, что в свою очередь является малоэффективным и значительно замедляет процесс обработки в условиях нынешней конкурентной среды. Также этот факт влечет за собой большое число ошибок. Все это существенно снижает эффективность работы предприятия.

Учет с использованием правильно спроектированной базы данных имеет свои особенности и радикально отличается от используемого. Использование базы данных облегчает учет, сокращая время, требующееся на оформление документов и обобщение накопленных данных для анализа хода деятельности, необходимого для управления ею. Отчеты о текущем состоянии, получаемые с помощью компьютера, можно получить и без него, но на расчеты уйдет много времени.

Программное обеспечение для работы с базами данных используется на персональных компьютерах уже довольно давно. К сожалению, эти программы либо были элементарными диспетчерами хранения данных и не имели средств разработки приложений, либо были настолько сложны и трудны, что даже хорошо разбирающиеся в компьютерах люди избегали работать с ними до тех пор, пока не получали полных, ориентированных на пользователя приложений.

Технология внутри машинной организации задается последовательностью реализуемых процедур – схем взаимосвязи программных модулей и информационных массивов. Такая схема представляет собой декомпозицию общего процесса решения задачи на отдельные процедуры преобразования массивов,

именуемыми модулями (это – ввод, контроль, перезапись информации с одного носителя на другой, сортировка, уплотнение данных, редактирование, накопление, вывод на печать и т.п.).

1.4 Анализ основных экономических показателей предприятия

Для наглядного отображения уровня доходов, расходов и прибыли предприятия построим соответствующие диаграммы. Так как предприятие начала свою деятельность с 2014 года, будут рассмотрены и проанализированы данные за три года. Рассмотрим диаграмму на рисунке 3, на которой представлены доходы в период с 2014 по 2016 г.

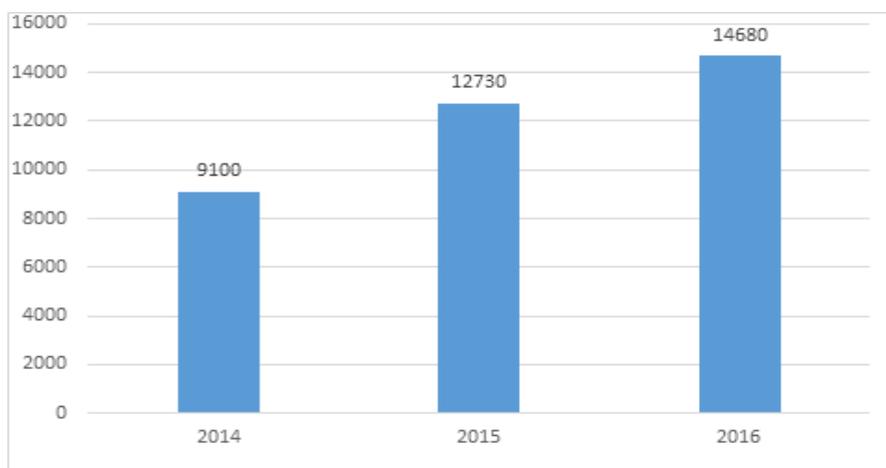


Рисунок 3 – Доходы ООО «Эрлант» за 2014 - 2016 гг. (тыс. руб.)

Анализируя полученную диаграмму можно определить, что уровень доходов за 2016 год является максимальным и на 15% выше уровня доходов за 2015 год, а по сравнению с 2014 г. превосходит на 22%.

Далее рассмотрим диаграмму, представленную на рисунке 4 – Расходы предприятия за три года, в период с 2014 по 2016 г.

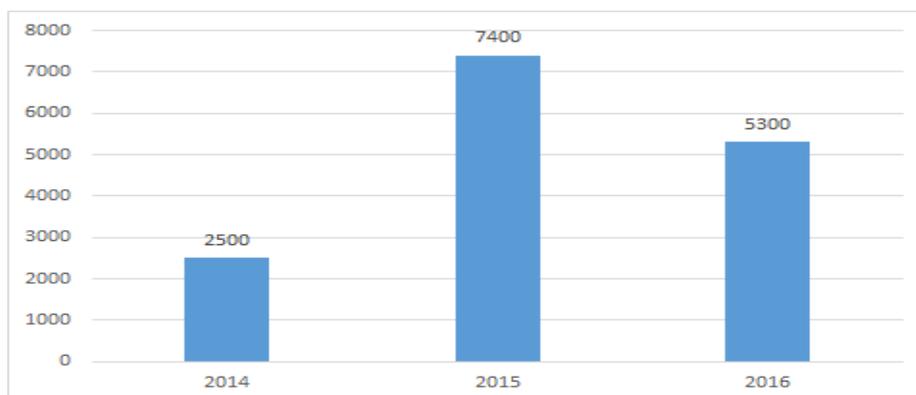


Рисунок 4 – Расходы ООО «Эрлант» за 2014 - 2016 гг. (тыс. руб.)

Анализируя диаграмму расходов, следует подчеркнуть, что расходы за 2015 год являются максимальными. За 2015 – 2016 гг. расходы держатся примерно на одном уровне.

Прибыль за каждый год составит:

2014 год: $9100 - 2500 = 6600$ тыс. руб.

2015 год: $12730 - 7400 = 5330$ тыс. руб.

2016 год: $14680 - 5300 = 9380$ тыс. руб.

Исходя из полученных значений, построим диаграмму прибыли (рисунок 5). Анализ диаграммы прибыли показывает, что первый год после открытия, является самым прибыльным. Самым не прибыльным является второй год, что связано с закупкой нового оборудования и расширением персонала. На третий год, предприятие стала увеличивать прибыль, так как все отделы были укомплектованы и не производилось больших технических затрат.

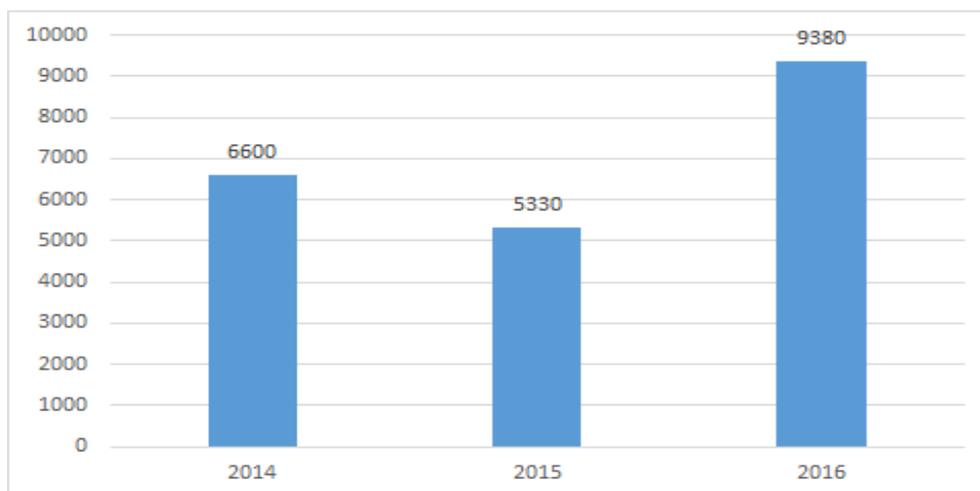


Рисунок 5 – Прибыль ООО «Эрлант» за 2014 - 2016 гг. (тыс. руб.)

Согласно предоставленным бухгалтерией ООО «Эрлант» данным, деятельность предприятия является прибыльной.

Таким образом, анализ экономических показателей ООО «Эрлант» показал, что его деятельность является прибыльной, предприятие развивается достаточно стабильно.

1.5 Документооборот предприятия

Документооборот – это движение документов в организации с момента их создания или получения до завершения исполнения или отправки.

В работе документооборот предприятия описан при помощи диаграммы потоков данных.

Диаграмма в нотации DFD описывает:

- функции обработки информации (работы);
- документы (стрелки), объекты, сотрудников или отделы, которые участвуют в обработке информации;
- внешние ссылки, которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
- таблицы для хранения документов (хранилища данных).

1.5.1 Внешний документооборот

Внешний документооборот представляет собой взаимодействие предприятия с внешними субъектами. Внешний документооборот компании представлен на рисунке 6.

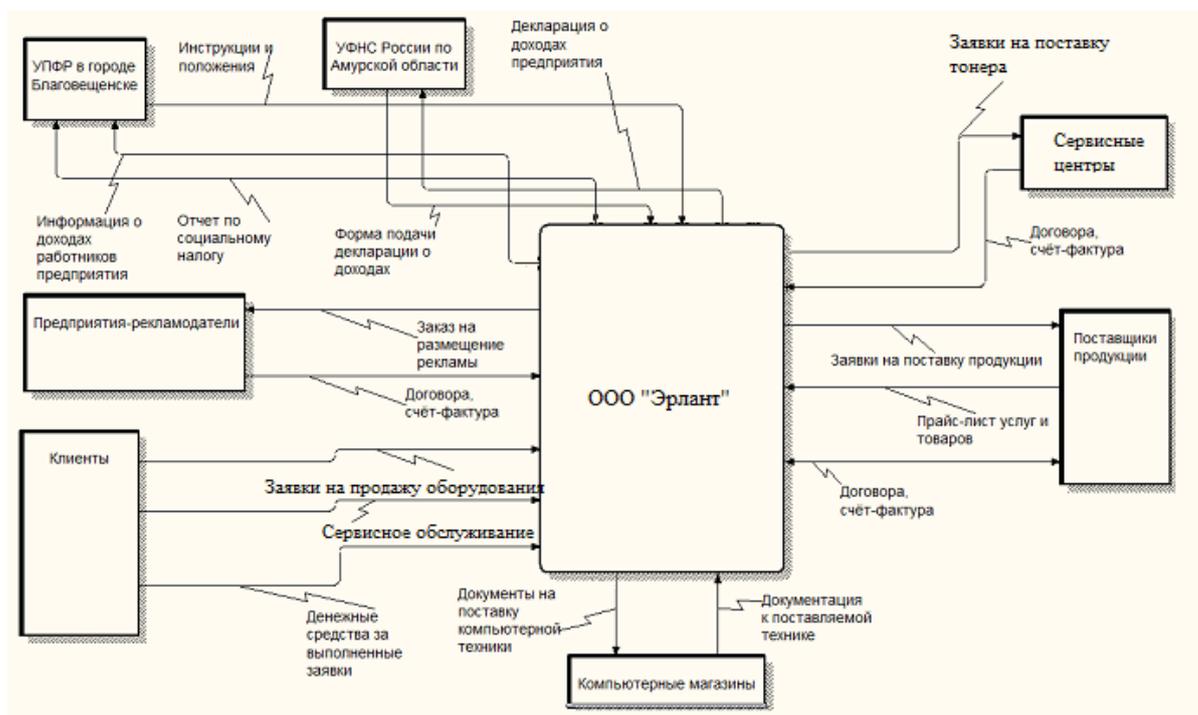


Рисунок 6 – Внешний документооборот ООО «Эрлант»

В ходе своей деятельности ООО «Эрлант» осуществляет взаимодействие с различными внешними организациями, государственными органами, различными субъектами. Основанием для взаимодействия с внешними контрагентами являются различные документы, регламентирующие деятельность предприятия

с тем или иным субъектом отношений, а также различные акты, постановления Российской Федерации, внутренние распорядки на предприятии и его вид деятельности.

В соответствии с рисунком 3 ООО «Эрлант» взаимодействует со следующими контрагентами:

- УПФР в городе Благовещенске;
- УФНС России по Амурской области;
- сервисные центры;
- поставщики продукции;
- компьютерные магазины;
- клиенты;
- предприятия-рекламодатели.

УПФР в городе Благовещенске регулирует взаимоотношения работодателя и работника, обеспечивая отчисления единого социального налога на заработную плату трудящегося, пополняющего денежные средства пенсионного фонда, из которого производятся пенсионные выплаты.

УФНС России по Амурской области обеспечивает отчисление предприятием всех налогов, контролирует деятельность предприятия, предоставляет форму подачи декларации о доходах.

Сервисные центры на основании договора с ООО «Эрлант» предоставляют тонер, для заправки картриджей.

Поставщики продукции на основании договора и заявок ООО «Эрлант» осуществляют отгрузку товара, своевременно обеспечивают торговый отдел каталогами товаров и услуг.

Компьютерные магазины предоставляют ООО «Эрлант» компьютерную технику, комплектующие и периферию на обслуживание расходных материалов.

Клиентами ООО «Эрлант» могут быть как физические лица, так и юридические. Клиенты предприятия подают заявки на покупку оборудования, а предприятие, в свою очередь, выполняет эти заявки.

Предприятия рекламодатели осуществляют рекламные компании, проводимые отделом маркетинга ООО «Эрлант».

1.5.2 Внутренний документооборот

Внутренний документооборот – это движение документов внутри организации с момента их создания или получения до завершения исполнения или отправления, то есть полный цикл жизни всего множества документов в организации до передачи их в архив или на уничтожение. Данный документооборот представлен на рисунке 7.

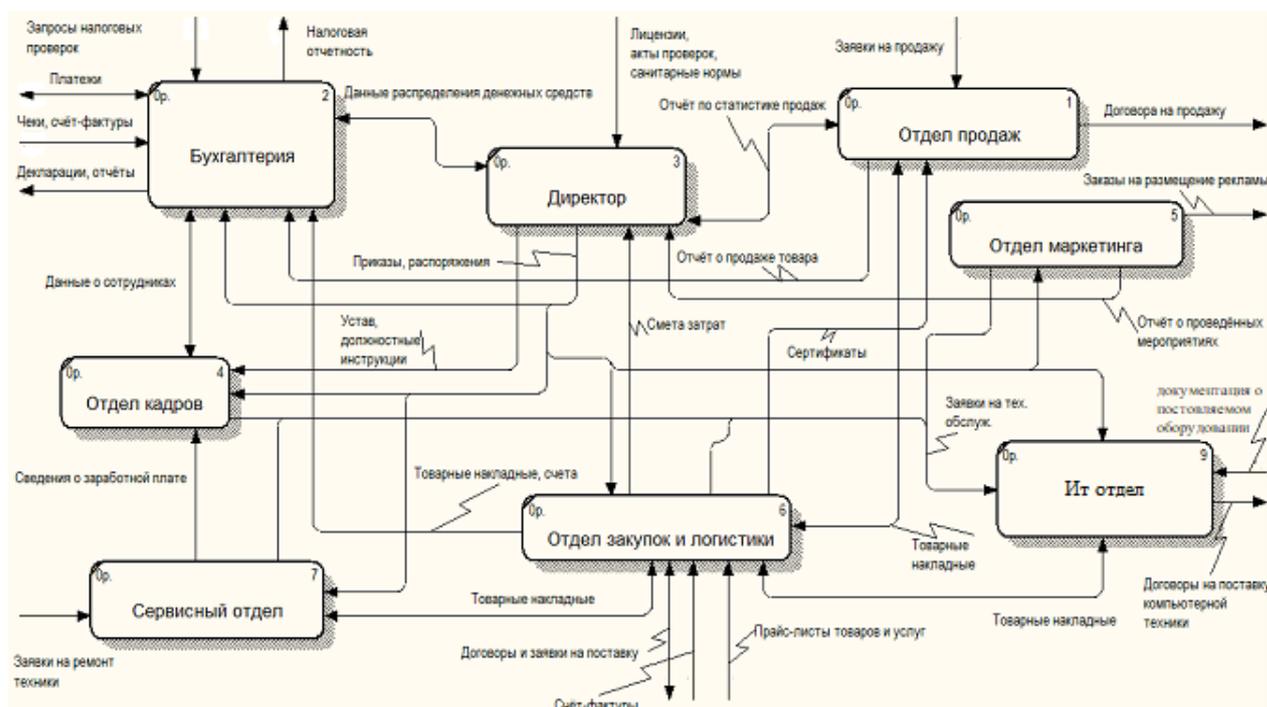


Рисунок 7 – Внутренний документооборот «Эрлант»

Все документы ООО «Эрлант» распределяются в соответствии с функциями и задачами подразделений и личного состава. Эти функции закреплены в уставах структурных единиц и в должностных инструкциях сотрудников.

К внутренней информации организации относятся:

- приказы и распоряжения руководителя по организации работы подразделений;
- отчеты и доклады, предоставляемые руководителю;
- результаты анализа деятельности;
- бухгалтерский учет и другая обязательная отчетность за различные пе-

риоды работы предприятия;

- счёт-фактуры, сметы;
- товарные накладные, сертификаты.

Все подразделения получают распоряжения, акты, приказы от руководителя, в ответ они представляют руководителю отчеты о проделанной работе.

Бухгалтерия получает указания и предписания из вышестоящих инстанций, осуществляет бухгалтерский и другие виды учета, направляет необходимую отчетность и разного рода документации в вышестоящие организации.

Законность документа подтверждается печатью и подписями уполномоченных лиц (для бумажного документа) и электронной подписью (для электронного документа).

Уничтожение документов допускается только после проведения проверки истечения нормативных сроков их хранения на основании надлежащим образом составленных и согласованных актов списания.

2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Назначение и цели создания системы

Разрабатываемая система предназначена автоматизации отдела информационных технологий, а именно для учета разрабатываемых на предприятии проектов, обслуживания и ремонта компьютерной и копировально-множительной техники предприятия.

Задачи, которые должны быть достигнуты при реализации системы:

- учет состояния вычислительной и копировально-множительной техники предприятия;
- прогнозирование оптимального количества запасных частей для восстановления вычислительной и копировально-множительной техники предприятия;
- планирование и контроль выполнения проектов;
- формирование рейтинга специалистов отдела.

Входными данными должны быть:

- по разрабатываемым проектам – наименование клиента, название проекта, описание проекта, дата начала работ, дата окончания работ, стоимость разрабатываемого проекта;
- данные о компьютерной и копировально-множительной техники (расходных материалах, запчастях) – наименование, производитель, поставщик, ответственное лицо, сведения о сервисе и ремонте.

Выходными данными должны быть:

- сотрудники, задействованные в каждом из проектов;
- история сервиса каждого компьютера;
- список всех комплектующих, находящихся на гарантии;
- комплектующие, которые необходимо заказать;
- комплектующие, которые необходимо заменить;
- список проектов по заказчикам;
- рейтинг сотрудников.

Все выходные данные должны иметь возможность выгрузки в MS Excel. Также должна быть возможность графического отображения выходных данных (в виде диаграммы).

Техническое задание на разработку системы приведено в приложении А.

2.2 Выбор среды разработки

Под средой разработки понимается набор программных средств, используемых программистами для разработки программного обеспечения.

В качестве среды разработки автоматизированной системы выбрана Delphi.

Общепризнанным является тот факт, что Delphi является универсальным средством для решения самых различных проблем программирования. Причем нельзя не отметить, что эта система сделала колоссальный шаг вперед по сравнению с остальными средствами визуального проектирования приложений. Delphi была первой системой ускоренной разработки приложений (RAD – Rapid Application Development), в которой удачно соединились средства визуального проектирования и оптимизирующий компилятор. Плюс к этому обширная библиотека компонентов, полная поддержка Internet, многоуровневая архитектура баз данных. Вот почему многие считают Delphi самой мощной системой разработки среди всех известных систем аналогичного назначения для Windows. Если Delphi и не является панацеей, то она очень сильно приближается к тому, чтобы именоваться таким образом.

На сегодняшний день на рынке имеется немало хороших средств разработки, позволяющих создавать приложения для архитектуры клиент/сервер. PowerBuilder, Visual Basic и другие подобные системы охватывают достаточно большие сектора рынка средств разработки баз данных. Однако, Delphi имеет ряд преимуществ.

Как уже говорилось выше, в Delphi удачно сочетаются средства визуального проектирования приложений и оптимизирующий компилятор, чего, к сожалению, нельзя сказать о других системах RAD. Наличие в системе компилятора или генерирование ею выполняемого машинного кода еще

не означает, что получаемый код является оптимальным. Такие системы, как PowerBuilder и Visual Basic, изначально создавались на основе концепции генерации псевдокода. При выполнении приложений, созданных с помощью этих систем, полученный псевдокод интерпретировался. Если сравнивать вышеупомянутые языки с языком Object Pascal, используемом в системе Delphi, то различие видно сразу. Object Pascal всегда был компилируемым языком, и при его разработке были соблюдены все требования, выполнение которых обязательно при компиляции и оптимизации. В итоге, Delphi является единственным полноценным средством промышленной разработки систем клиент/сервер. Сравнение Delphi с Visual Basic или PowerBuilder подобно сравнению современного компилятора языка C++ с компилятором Clipper времен господства DOS. Только успех Delphi подвиг многих производителей средств разработки приложений клиент/сервер добавить в свои продукты технологию генерации машинного кода.

В случае Delphi слова «оптимизирующий компилятор» не означают «медленный компилятор». В последней версии продукта представлен лучший компилятор языка Pascal компании Borland, которая уже на протяжении многих лет удерживает пальму первенства в этой области. Компиляторы Pascal этой компании снискали заслуженную славу за генерацию выполняемого кода, который экономно использует ресурсы компьютера и одновременно обладает высокой производительностью. Компилятор Object Pascal, используемый в Delphi, не является исключением. Более того, генератор кода Delphi - это тот самый генератор кода, который применяется компанией Borland в ее компиляторах с языка C++. Таким образом, используя Delphi, можно достичь скорости программ, написанных на языке C++, избежав при этом сложностей этого языка программирования.

Помимо эффективности выполняемого кода, каждому разработчику необходимо средство, которое было бы, с одной стороны, достаточно мощным и гибким, чтобы выполнить любую стоящую перед ними задачу, а, с другой стороны - было достаточно простым и удобным в работе. Такая система должна

быть построена на принципах объектно-ориентированного программирования, и вместе с тем, должна позволять применять, в случае необходимости, ассемблер. Она должна генерировать не только исполняемые файлы в формате EXE, но и создавать файлы DLL и драйверов устройств. Наконец, она совмещать в себе функции среды для быстрой разработки баз данных, которая не вынуждала бы их каждый раз при программировании спускаться до ядра СУБД.

2.3 Функциональная структура системы

Функциональная структура автоматизированной информационной системы представлена на рисунке 8.

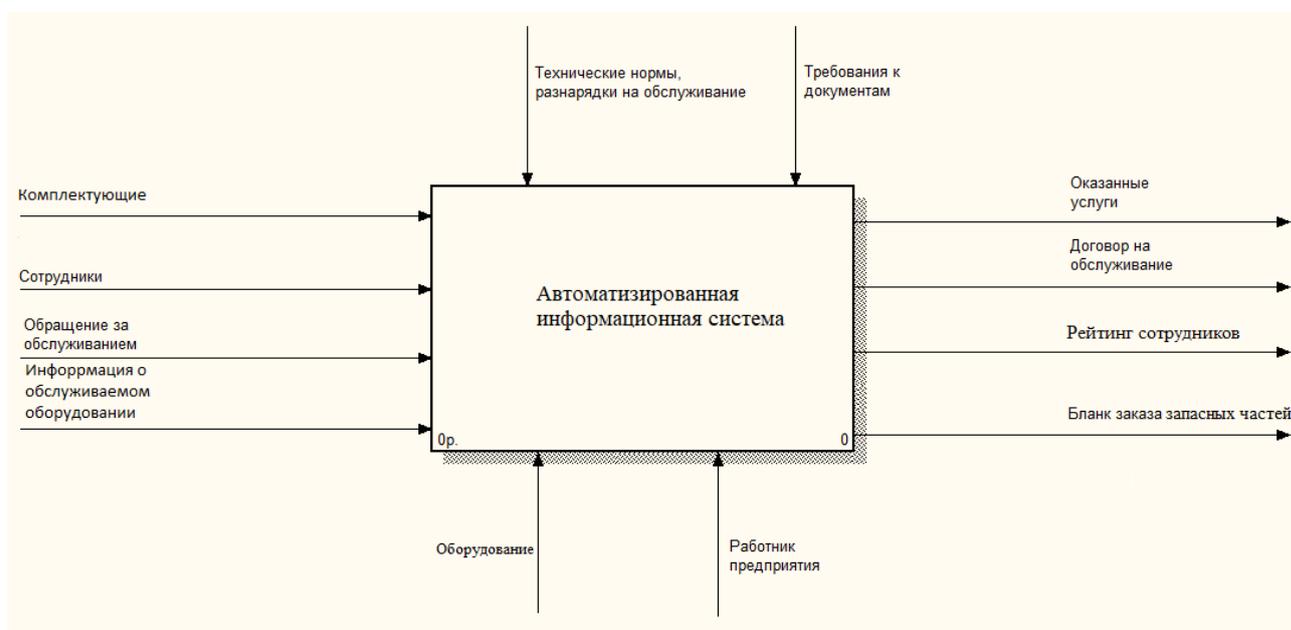


Рисунок 8 – Функциональная структура системы

Из функциональной структуры системы видно, что основными входящими потоками для системы является информации о клиентах, сотрудниках, обслуживаемой технике, имеющихся комплектующих, а также обращение за обслуживанием.

К основным выходящим потокам относятся: оказанные услуги, договор на обслуживание, расписание работ, бланк заработной платы, а также бланк заказа запасных частей оборудования.

На рисунке 9 представлена декомпозиция функциональной структуры.

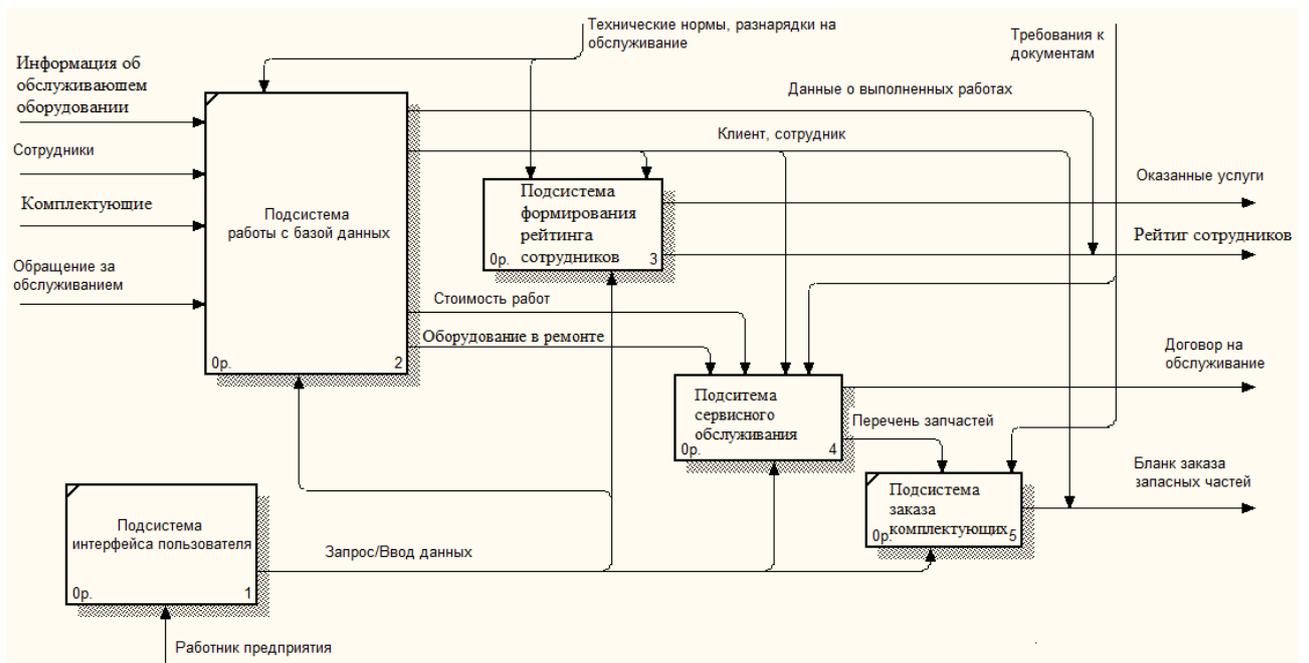


Рисунок 9 – Декомпозиция функциональной структуры системы

Преимуществом данной системы является то, что она построена с использованием принципов ООП. Использование данного метода программирования обладает тремя главными преимуществами:

- сама разработка проста для понимания – ООП позволяет мыслить категориями повседневных объектов;
- разработанная система надежна и проста для сопровождения – правильное проектирование обеспечивает простоту расширения и модификации объектно-ориентированных программ. Модульная структура позволяет вносить независимые изменения в разные части программы, сводя к минимуму риск ошибок программирования;
- ООП ускоряет цикл разработки – модульность и здесь играет важную роль, поскольку различные компоненты объектно-ориентированных программ можно легко использовать в других программах, что уменьшает избыточность кода и снижает риск внесения ошибок при копировании.

2.4 Модули системы

Объектно-ориентированное программирование заметно повышает эффективность труда программистов и позволяет создавать более мощные, масштабируемые и эффективные приложения. Многие преимущества ООП обусловле-

ны одним из его фундаментальных принципов – инкапсуляцией. Инкапсуляцией называется включение различных мелких элементов в более крупный объект, в результате чего программист работает непосредственно с этим объектом. Это приводит к упрощению программы, поскольку из нее исключаются второстепенные детали. В ООП эта возможность обеспечивается классами, объектами и различными средствами выражения иерархических связей между ними. Классы образуют синтаксическую базу объектно-ориентированного программирования. Их можно рассматривать как своего рода «контейнеры» для логически связанных данных и функций.

Класс представляет собой шаблон, по которому создаются конкретные экземпляры, используемые в программе. Экземпляры классов называются объектами.

В разработанной системе классами представлены функциональные подсистемы или модули. Так, например, модуль работы с базой данных представлен классом «work_base». Для удобства за названия классов примем названия соответствующих модулей.

В каждом из классов существуют два типа методов – методы для обработки данных внутри класса и методы для взаимодействия с другими классами.

Первый тип методов необходим по большей части для удобства разработки, а также для простоты понимания исходного кода. Так, при выполнении sql-запроса к базе данных сначала нужно функцией «mysql_query» передать запрос в базу данных, после выполнения которого, данная функция запишет результат выполнения запроса во временную таблицу и указатель, на которую вернет с возвращаемым значением функции. После этого функцией «mysql_fetch_array» можно построчно считать данные из таблицы. Т.е. для выполнения одного sql-запроса необходимо как минимум использование двух функций и одного цикла, не считая проверок на корректность данных, проверок существования подключения к базе данных и проверок, прав доступа к базе. Но эту процедуру можно упростить, создав один метод, в котором выполняются все необходимые операции.

Второй тип методов, методы взаимосвязи, которые на основе переданных параметров формируют и выполняют запросы к базе. Получив ответ от базы данных, выбранная информация формируется в удобном для обработки виде и передается с возвращаемым значением функции. В данных методах используются как простейшие запросы на выборку и добавление данных, например, поисковые в которых передаются имя, фамилия и номер группы, а возвращаются личные данные. Так и сложные составные запросы, необходимы для выполнения различных операций, по формированию иерархии групп пользователей и тем или при выборке вопросов тестирования. Второй тип вопросов необходим по большей части для ускорения работы системы, так как один большой составной запрос к базе данных выполнится намного быстрее, чем несколько простых. Это связано с тем, что при выполнении запроса к БД все записи последовательно считываются с носителя информации и сравниваются на соответствие с запросом, который в этот момент находится в оперативной памяти. Время считывания записи с жесткого диска или другого подобного носителя и время занесения ее во временную таблицу зачастую больше чем время сравнения записи. Поэтому нецелесообразно несколько раз делать простейшие запросы к базе данных, которые в свою очередь будут заново проходить все записи в базе и выявлять соответствия, особенно если записей большое количество.

Основным классом в разработанной системе можно считать класс работы с базой данных, это связано с тем, что все классы напрямую зависят от модуля взаимодействия с базой данных, иными словами без него система будет полностью неработоспособной. Действительно данный класс в первую очередь инициализируется при запуске системы, он проверяет: наличие подключения к серверу; наличие на сервере необходимой базы данных; возможность подключения к базе и целостность данных.

2.5 Проектирование базы данных

При проектировании структуры новой базы данных определяют сущности (объекты, явления) предметной области, которые должны найти свое отражение в базе данных.

Объект – это такая абстракция множества предметов реального мира, что все экземпляры этого объекта имеют одни и те же характеристики и подчиняются одним и тем же правилам поведения. Объекты обладают определенными свойствами – атрибутами.

Атрибут – это абстракция одной характеристики объекта. Каждый атрибут имеет имя и может получать значения из некоторого множества допустимых значений. Как правило, каждому объекту в базе данных соответствует таблица, а его атрибутам – поля этой таблицы.

В результате анализа были выделены 11 объектов, которые описывают данную предметную область.

Сущность «Сотрудник». Она включает в себя основные сведения о поставщиках и Сотрудниках организации. Атрибутами сущности являются «id Сотрудника» (первичный ключ), «Наименование», «Контактная информация».

Сущность «Комплект». Она включает в себя сведения о комплектующих и расходных материалах. Атрибутами сущности являются «id комплект» (первичный ключ), «Модель», «id сотрудника» (внешний ключ), «id типа» (внешний ключ), «id производителя» (внешний ключ), «id Сотрудника» (внешний ключ), «Дата выпуска», «Стоимость» и «Гарантия до».

Сущность «Сотрудники предприятия». Она включает в себя сведения о сотрудниках организации и закрепленных за ними компьютерах. Атрибутами сущности являются «id сотрудника» (первичный ключ), «ФИО сотрудника», «id отдела» (внешний ключ), «Должность», «Функции», «Имя компьютера», «IP», «Дата ввода», «Инвентарный» и «Описание».

Сущность «Отделы», которая включает в себя сведения об отделах организации. Атрибутами являются «ID отд» (первичный ключ), «Название отдела», «ФИО начальника», «Описание».

Сущность «Типы», которая содержит сведения о типах комплектующих и расходных материалов. Атрибутами являются «Id типа» (первичный ключ), «Тип» и «Характеристика».

Сущность «Осмотр», которая включает в себя сведения о сервисном об-

служивании комплектующих. Атрибутами сущности являются «id» (первичный ключ), «Дата осмотра», «id комп» (внешний ключ), «id мастера» (внешний ключ), «Неисправность», «Замена», «Гарантия» и «Заказ».

Сущность «Проекты». Она включает в себя основные сведения о проектах, разрабатываемых в организации. Атрибутами сущности являются «id проекта» (первичный ключ), «id Сотрудника» (внешний ключ), «Название проекта», «Описание проекта», «Начало», «Окончание», «Стоимость» и «Состояние».

Сущность «Мастера» включает в себя сведения о мастерах, производящих сервисное обслуживание. Атрибутами сущности являются «id мастера» (первичный ключ), «ФИО мастера», «Квалификация».

Сущность «Работы», которая включает в себя сведения о выполняемых по проектам работах. Атрибутами являются «id работы» (первичный ключ), «Дата», «Характеристика», «id выполнения» (внешний ключ).

Сущность «Выполнение», которая содержит сведения о выполнении работ по проекту. Атрибутами являются «id выполн» (первичный ключ), «id проекта» (внешний ключ), «id сотрудника» (внешний ключ) и «Характеристика».

Сущность «Производители», которая включает в себя сведения о производителях комплектующих. Атрибутами сущности являются «Id производит» (первичный ключ), «Производитель» и «Страна».

Описание атрибутов каждой сущности приведено в таблицах 1 – 11.

Сущность «Отделы» предназначена для хранения сведений об отделах организации. Описание полей данной сущности представлено в таблице 1.

Таблица 1 – Структура сущности «Отделы»

Название столбца	Описание
id	Первичный ключ. Обеспечивает уникальность каждой записи.
Название отдела	Название отдела
ФИО начальника	ФИО начальника отдела
Описание	Характеристика отдела

Сущности «Сотрудники предприятия» содержит информацию о людях, работающих в каждом из отделов. Описание полей данной сущности представ-

лено в таблице 2.

Таблица 2 – Структура сущности «Сотрудники предприятия»

Название столбца	Описание
id	Первичный ключ. Обеспечивает уникальность каждой записи.
id отдела	Внешний ключ. Номер отдела.
ФИО сотрудника	ФИО сотрудника отдела
Должность	Занимаемая должность
Функции	Выполняемые сотрудником функции
Имя компьютера	Сетевое имя компьютера
IP	IP адрес компьютера
Дата ввода	Дата ввода в эксплуатацию
Инвентарный	Инвентарный номер компьютера
Описание	Описание компьютера

В сущности, «Комплект» содержатся сведения о комплектующих каждого из компьютеров. Описание сущности представлено в таблице 3.

Таблица 3 – Структура сущности «Комплект»

Название столбца	Описание
id	Первичный ключ. Обеспечивает уникальность каждой записи.
id сотрудника	Внешний ключ. Номер сотрудника (компьютера).
id типа	Внешний ключ. Номер типа комплектующего.
Модель	Модель комплектующего
id производителя	Внешний ключ. Номер производителя
id поставщика	Внешний ключ. Номер поставщика.
Дата выпуска	Дата выпуска комплектующего
Стоимость	Стоимость комплектующего
Гарантия до	Дата истечения гарантии

В сущности, «Осмотр» содержатся сведения об осмотрах, комплектующих каждого из компьютеров. Описание сущности представлено в таблице 4.

Таблица 4 – Структура сущности «Осмотр»

Название столбца	Описание
id	Первичный ключ. Обеспечивает уникальность каждой записи.
id комплектующего	Внешний ключ. Номер производителя
Дата осмотра	Дата осмотра
id мастера	Номер мастера, производившего осмотр.
Неисправность	Выявленная неисправность
Замена	Признак необходимости замены
Гарантия	Признак возможности замены по гарантии
Заказ	Признак необходимости заказа комплектующего

В сущности, «Типы» содержатся наименования типов (категорий) комплектующих. Описание сущности представлено в таблице 5.

Таблица 5 – Структура сущности «Типы»

Название столбца	Описание
id	Первичный ключ. Обеспечивает уникальность каждой записи.
Тип	Наименование типа (категории) комплектующего
Характеристика	Характеристика типа комплектующего

В сущности, «Производители» содержатся наименования о производителях комплектующих. Описание сущности представлено в таблице 6.

Таблица 6 – Структура таблицы «Производители»

Название столбца	Описание
id	Первичный ключ. Обеспечивает уникальность каждой записи.
Производитель	Наименование производителя комплектующего
Страна	Страна-производитель комплектующего

В сущности, «Проекты» содержатся сведения о проектах, выполняемых организацией. Описание сущности представлено в таблице 7.

Таблица 7 – Структура таблицы «Проекты»

Название столбца	Описание
id	Первичный ключ. Обеспечивает уникальность каждой записи.
Название проекта	Название проекта
id Сотрудника	Номер заказчика
Описание проекта	Характеристика выполняемого проекта.
Начало	Дата начала работ по проекту
Окончание	Дата окончания работ по проекту
Стоимость	Стоимость выполнения проекта
Состояние	Состояние работ над проектом

В сущности, «Выполнение» содержатся сведения о сотрудниках, задействованных в выполнении проекта. Описание сущности представлено в таблице 8.

Таблица 8 – Структура сущности «Выполнение»

Название столбца	Описание
id	Первичный ключ. Обеспечивает уникальность каждой записи.
id сотрудника	Внешний ключ. Номер сотрудника (компьютера).
id проекта	Внешний ключ. Номер проекта.
Характеристика	Роль сотрудника в проекте

В сущности, «Работы» содержатся данные о работе сотрудника по проекту. Описание сущности представлено в таблице 9.

Таблица 9 – Структура таблицы «Работы»

Название столбца	Описание
id	Первичный ключ. Обеспечивает уникальность каждой записи.
Id выполнения	Внешний ключ. Номер выполняемого сотрудником проекта.
Дата	Дата выполнения работы
Характеристика	Характеристика выполняемой работы

В сущности, «Поставщики» содержатся данные о поставщиках и заказчиках предприятия. Описание полей представлено в таблице 10.

Таблица 10 – Структура сущности «Поставщики»

Название столбца	Описание
id	Первичный ключ. Обеспечивает уникальность каждой записи.
Наименование	Наименование поставщика/заказчика сотрудником проекта.
ИНН	ИНН Сотрудника
КПП	КПП Сотрудника
Контактная информация	Контактная информация о Сотруднике
Банк	Банковские реквизиты Сотрудника
Руководитель	Руководитель организации

В сущности, «Мастера» содержатся данные о мастерах, производящих осмотр и ремонт компьютеров. Описание сущности представлено в таблице 11.

Таблица 11 – Структура сущности «Поставщики»

Название столбца	Описание
Id	Первичный ключ. Обеспечивает уникальность каждой записи.
ФИО мастера	ФИО мастера
Квалификация	Квалификация мастера

2.5.1 Инфологическое проектирование

Между объектами предметной области существуют связи, которые должны быть отражены в виде связей между объектами инфологической модели. Связь является логическим соотношением между сущностями. Графически связь обозначается линией, соединяющей связываемые объекты. В каждом направлении связи можно выделить главный объект, от которого идет связь, и

подчиненный.

Различают идентифицирующую связь и не идентифицирующую связь. При установлении не идентифицирующей связи дочерняя сущность остается независимой. Экземпляр сущности родителя может существовать безотносительно к какому-либо экземпляру дочерней сущности. При идентифицирующей связи экземпляр подчиненной сущности зависит от родительской сущности и не может существовать без экземпляра родительской сущности.

Целью данного этапа является построение реляционной логической модели. Реляционная логическая модель представляет собой совокупность нормализованных отношений, в которых реализованы связи между объектами предметной области и выполнены все преобразования, необходимые для ее эффективной реализации в среде конкретной СУБД.

1) Связь «Мастера – Осмотр» является связью типа «один–ко–многим». При отображении ключ порожденной сущности добавляется в исходную сущность. Исходной сущностью является сущность «Мастера», порожденной – «Осмотр». Связь показана на рисунке 10, на рисунке 11 приведены итоговые отношения.



Рисунок 10 – Связь «Мастера – Осмотр»



Рисунок 11 – Результат анализа связи «Мастера – Осмотр»

2) Связь «Комплект – Осмотр» является связью типа «один–ко–многим». При отображении ключ порожденной сущности добавляется в исходную сущность. Исходной сущностью является сущность «Комплект», порожденной –

«Осмотр». Связь показана на рисунке 12, на рисунке 13 приведены итоговые отношения.



Рисунок 12 – Связь «Комплект – Осмотр»

Сущность – «Комплект»

<u>id комплект</u>	Модель	Дата выпуска	Стоимость	Гарантия до
--------------------	--------	--------------	-----------	-------------

Сущность – «Осмотр»

<u>id</u>	<i>id комплект</i>	Дата осмотра	Id мастер	Дата осмотра	Неисправность	Замена
Гарантия		Заказ				

Рисунок 13 – Результат анализа связи «Комплект – Осмотр»

3) Связь «Производители – Комплект» является связью типа «один–многим». При отображении ключ порожденной сущности добавляется в исходную сущность. Исходной сущностью является сущность «Производители», порожденной – «Комплект». Связь показана на рисунке 14, на рисунке 15 приведены итоговые отношения.



Рисунок 14 – Связь «Производители – Комплект»

Сущность – «Производители»

<u>id производителя</u>	Производитель	Страна
-------------------------	---------------	--------

Сущность – «Комплект»

<u>id комплект</u>	<i>Id производителя</i>	Модель	Дата выпуска	Стоимость	Гарантия до
--------------------	-------------------------	--------	--------------	-----------	-------------

Рисунок 15 – Результат анализа связи «Производители – Комплект»

4) Связь «Типы – Комплект» является связью типа «один–ко–многим». При отображении ключ порожденной сущности добавляется в исходную сущность. Исходной сущностью является сущность «Жители», порожденной – «Льготы». Связь показана на рисунке 15, на рисунке 16 приведены итоговые отношения.



Рисунок 16 – Связь «Типы – Комплект»



Рисунок 17 – Результат анализа связи «Типы – Комплект»

5) Связь «Клиенты – Комплект» является связью типа «один–ко–многим». При отображении ключ порожденной сущности добавляется в исходную сущность. Исходной сущностью является сущность «Клиенты», порожденной – «Комплект». Связь показана на рисунке 18, на рисунке 19 приведены итоговые отношения.

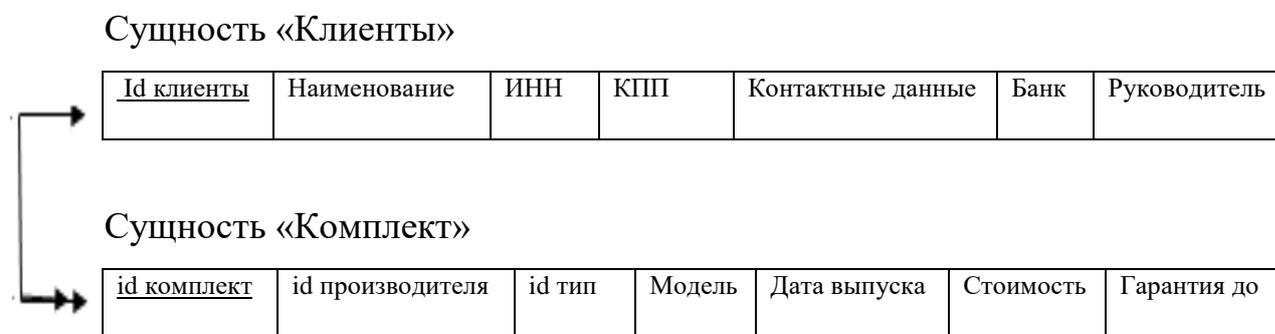


Рисунок 18 – Связь «Клиенты – Комплект»

Сущность «Клиенты»

<u>Id клиенты</u>	Наименование	ИНН	КПП	Контактные данные	Банк	Руководитель
-------------------	--------------	-----	-----	-------------------	------	--------------

Сущность «Комплект»

<u>id комплект</u>	id производителя	<i>id клиенты</i>	<u>id тип</u>	Модель	Дата выпуска	Стоимость
Гарантия до						

Рисунок 19 – Результат анализа связи «Клиенты – Комплект»

б) Связь «Сотрудники – Комплект» является связью типа «один–ко–одному». При отображении ключ порожденной сущности добавляется в исходную сущность. Исходной сущностью является сущность «Сотрудники», порожденной – «Комплект». Связь показана на рисунке 20. на рисунке 21 приведены итоговые отношения.

Сущность «Сотрудники»

<u>Id сотрудника</u>	ФИО Сотрудника	Должность	Функции	Имя компьютера	IP	Дата ввода
Инвентарный	Описание					

Сущность «Комплект»

<u>id комплект</u>	id производителя	<i>id клиенты</i>	id тип	Модель	Дата выпуска	Стоимость	Гарантия до
--------------------	------------------	-------------------	--------	--------	--------------	-----------	-------------

Рисунок 20 – Связь «Сотрудники – Комплект»

Сущность «Сотрудники»

<u>id сотрудника</u>	ФИО Сотрудника	Должность	Функции	Имя компьютера	IP	Дата ввода
Инвентарный	Описание					

Сущность «Комплект»

<u>id комплект</u>	id производителя	<i>id клиенты</i>	id тип	Модель	Дата выпуска	Стоимость	Гарантия до
<i>id сотрудника</i>							

Рисунок 21 – Связь «Сотрудники – Комплект»

7) Связь «Отделы – Сотрудники» является связью типа «один–ко–многим». При отображении ключ порожденной сущности добавляется в исходную сущность. Исходной сущностью является сущность «Отделы», порожден-

ной – «Сотрудники». Связь показана на рисунке 22, на рисунке 23 приведены итоговые отношения.



Рисунок 22 – Связь «Отделы – Сотрудники»

Сущность «Отделы»

<u>id отдела</u>	Название отдела	ФИО начальника	Описание
------------------	-----------------	----------------	----------

Сущность «Сотрудники»

<u>id сотрудника</u>	ФИО сотрудника	Должность	Функции	Имя компьютера	IP	Дата ввода
Инвентарный	Описание		<i>id отдела</i>			

Рисунок 23 – Результат анализа связи «Отделы – Сотрудники»

8) Связь «Сотрудники – Выполнение» является связью типа «один–многим». При отображении ключ порожденной сущности добавляется в исходную сущность. Исходной сущностью является сущность «Сотрудники», порожденной – «Выполнение». Связь показана на рисунке 24, на рисунке 25 приведены итоговые отношения.



Рисунок 24 – Связь «Сотрудники – Выполнение»

Сущность «Сотрудники»

<u>id сотрудника</u>	ФИО Сотрудника	Должность	Функции	Имя компьютера	IP	Дата ввода
Инвентарный	Описание		id отдела			

Сущность «Выполнение»

<u>id выполнения</u>	<u>id сотрудника</u>	Характеристика
----------------------	----------------------	----------------

Рисунок 25 – Результат анализа связи «Сотрудники – Выполнение»

9) Связь «Клиенты – Проекты» является связью типа «один–ко–многим». При отображении ключ порожденной сущности добавляется в исходную сущность. Исходной сущностью является сущность «Клиенты», порожденной – «Проекты». Связь показана на рисунке 26, на рисунке 27 приведены итоговые отношения.

Сущность «Клиенты»

<u>Id клиенты</u>	Наименование	ИНН	КПП	Контактные данные	Банк	Руководитель
-------------------	--------------	-----	-----	-------------------	------	--------------

Сущность «Проекты»

<u>id проекта</u>	Название проекта	Описание проекта	Начало	Окончание
Стоимость	Состояние			

Рисунок 26 – Связь «Клиенты – Проекты»

Сущность «Клиенты»

<u>Id клиенты</u>	Наименование	ИНН	КПП	Контактные данные	Банк	Руководитель
-------------------	--------------	-----	-----	-------------------	------	--------------

Сущность «Проекты»

<u>id проекта</u>	<u>id клиенты</u>	Название проекта	Описание проекта	Начало	Окончание
Стоимость	Состояние				

Рисунок 27 – Результат анализа связи «Клиенты – Проекты»

10) Связь «Проекты – Выполнение» является связью типа «один–ко–многим». При отображении ключ порожденной сущности добавляется в исходную сущность. Исходной сущностью является сущность «Проекты», порожденной – «Выполнение».

денной – «Выполнение». Связь показана на рисунке 28, на рисунке 29 приведены итоговые отношения.



Рисунок 28 – Связь «Проекты – Выполнение»

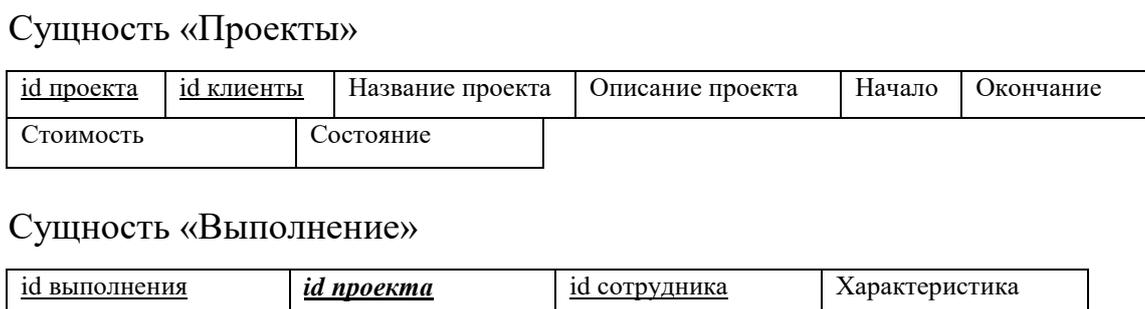


Рисунок 29– Результат анализа связи «Клиенты – Проекты»

11) Связь «Выполнение – Работы» является связью типа «один–ко–многим». При отображении ключ порожденной сущности добавляется в исходную сущность. Исходной сущностью является сущность «Выполнение», порожденной – «Работы». Связь показана на рисунке 30, на рисунке 31 приведены итоговые отношения.

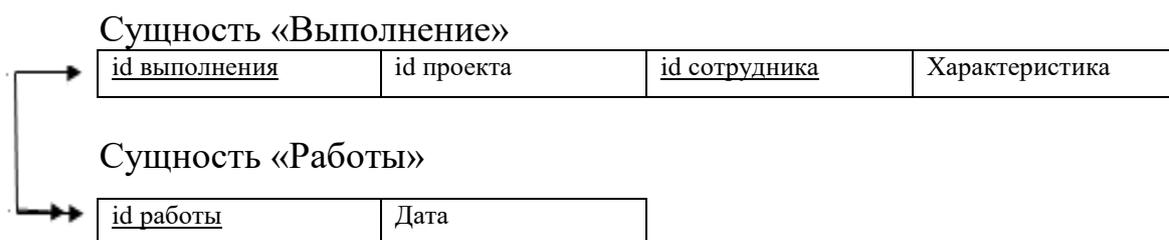


Рисунок 30 – Связь «Клиенты – Проекты»

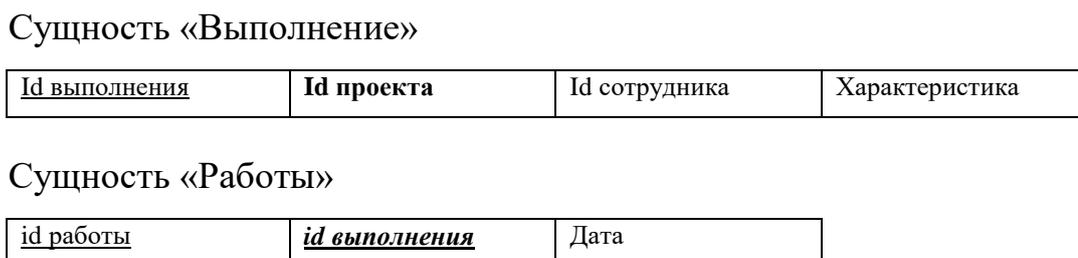


Рисунок 31 – Результат анализа связи «Клиенты – Проекты»

скольку все их атрибуты являются неделимыми.

Отношение находится во второй нормальной форме (2НФ), если оно находится в первой нормальной форме (1НФ), и каждый не ключевой атрибут функционально полно зависит от составного ключа. Полная зависимость означает, что не ключевой атрибут должен зависеть от всех ключевых атрибутов, а не от части. Те атрибуты, которые зависят только от части первичного ключа, должны быть выделены в отдельные таблицы. В отношениях, находящихся во второй нормальной форме могут присутствовать транзитивные зависимости, которые приводят к избыточности и аномалиям, например, аномалиям обновления. Такие транзитивные зависимости приводят к повторению атрибутов и аномалиям обновления при изменении значений какого-либо из этих атрибутов.

Эти недостатки устраняются в третьей нормальной форме. Отношение находится в третьей нормальной форме (3НФ), если оно находится во второй нормальной форме (2НФ), и в нем отсутствуют транзитивные зависимости не ключевых атрибутов от ключа, то есть не должно быть зависимостей между не ключевыми атрибутами.

Для приведения отношений в третью нормальную форму необходимо найти в нем цепочки транзитивных зависимостей и выделить отдельные отношения из атрибутов, замыкающих цепочку транзитивных зависимостей. Нормализация отношений производится с помощью неформальной декомпозиции.

Таким образом, требуется проверить, находятся ли отношения в третьей нормальной форме (нахождение в которой подразумевает и нахождение во всех остальных), и, если нет – привести все отношения схемы в нормальную форму путем декомпозиции.

Например, сущность «Выполнение» имеет составной первичный ключ, состоящий из трех атрибутов: «id выполнение», «id проекта» и «id сотрудника». Не ключевые атрибуты сущности функционально полно зависят от ключа, транзитивные зависимости между ними отсутствуют. Следовательно, сущность «Выполнение» находится в третьей нормальной форме.

Сущность «Осмотр» также находится в третьей нормальной форме, по-

сколькx имеет составной первичный ключ, в ней также отсутствуют транзитивные зависимости между атрибутами, не ключевые атрибуты («Дата осмотра», «Неисправность», «Замена», «Гарантия», «Заказ») функционально полно зависят от ключевых атрибутов.

Сущности «Сотрудники предприятия» и «Комплекующие» находятся в третьей нормальной форме и удовлетворяют всем предъявляемым требованиям.

Итак, при проектировании базы данных была проведена нормализация отношений до третьей нормальной формы, т.е. были устранены не ключевые атрибуты, не зависящие от ключа. Таким образом, все не ключевые атрибуты функционально полно зависят от ключа и отсутствуют транзитивные зависимости.

Проведенная проверка инфологической модели предметной области показала отсутствие повторяющихся групп и транзитивных зависимостей, а также функционально полную зависимость не ключевых атрибутов от составного ключа.

Это позволяет сделать вывод, что отношения находятся в третьей нормальной форме – в этой форме устраняются большинство аномалий и избыточность.

Тем не менее для сущности «Сотрудник» была выполнена денормализация. В данном случае атрибут «Контактная информация» не является атомарным. Однако, поскольку все элементы данного атрибута всегда используются вместе, то можно сделать вывод об обоснованности этой денормализации.

Также денормализация была проведена для сущностей «Сотрудника» и «Мастера», в которых атрибуты «ФИО» не являются неделимыми, но необходимость разбиения этого атрибута не приведет к повышению эффективности работы.

2.5.3 Физическое проектирование

Физическая модель содержит всю информацию, необходимую для реализации конкретной базы данных. Была произведена генерация схемы физической базы данных для модели предметной области в среде ERWin

путем выбора целевой СУБД FirebirdSQL. При этом сущности логической модели становятся таблицами базы данных, а атрибуты – полями таблиц.

Физическая модель информационной системы АСУ отдела информационных технологий представлена на рисунке 33.

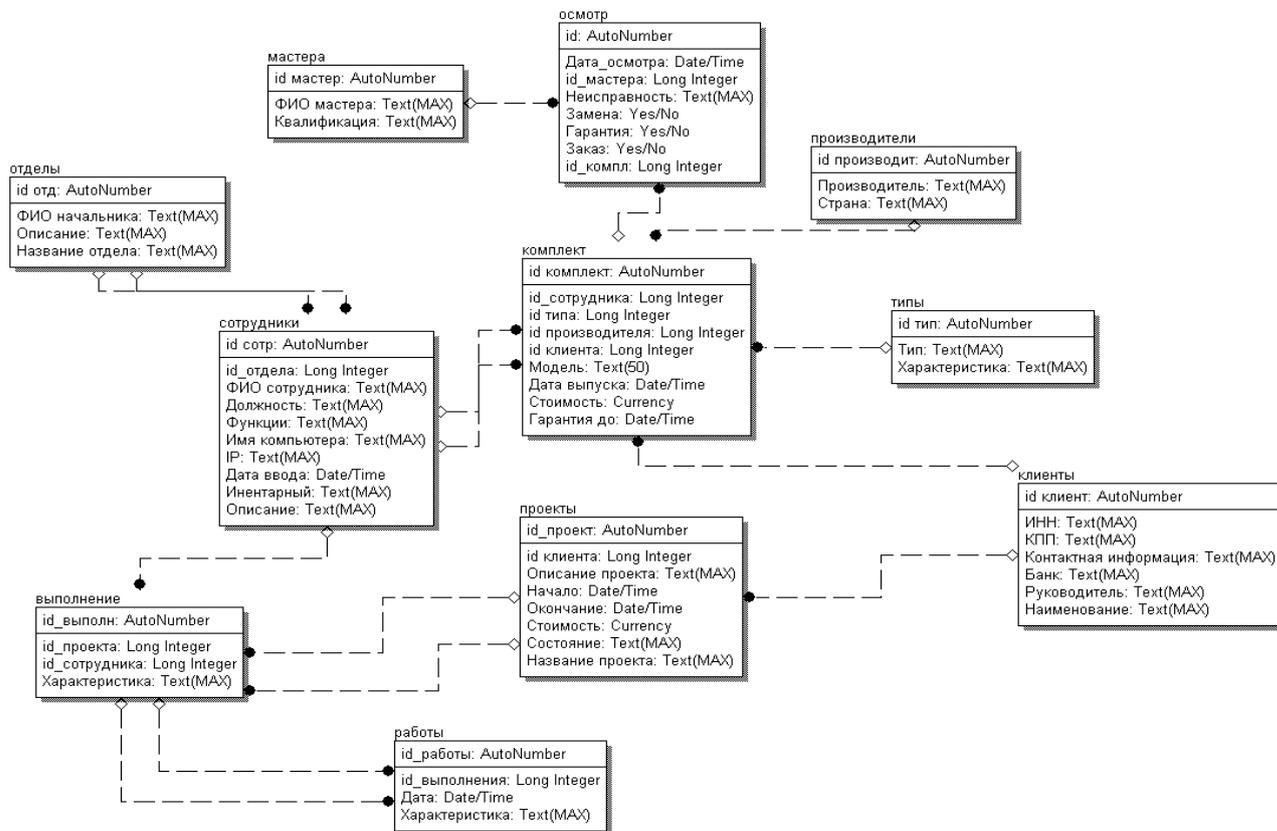


Рисунок 33 – Физическая схема модели

В процессе создания физической модели данных были скорректированы некоторые типы данных значений полей в соответствии с реализацией базы данных. Физические свойства полей приведены в таблицах 12-23.

Таблица 12 – Структура таблицы «Отделы»

Название столбца	Тип данных	Длина	Описание
Id	int	-	Первичный ключ. Обеспечивает уникальность каждой записи.
Название отдела	varchar	MAX	Название отдела
ФИО начальника	varchar	MAX	ФИО начальника отдела
Описание	varchar	MAX	Характеристика отдела

Таблица «Сотрудники предприятия» содержит информацию о людях, работающих в каждом из отделов. Описание полей данной таблицы представлено в таблице 13.

Таблица 13 – Структура таблицы «Сотрудники предприятия»

Название столбца	Тип данных	Длина	Описание
id	int	-	Первичный ключ. Обеспечивает уникальность каждой записи.
id отдела	int	-	Внешний ключ. Номер отдела.
ФИО сотрудника	varchar	MAX	ФИО сотрудника отдела
Должность	varchar	MAX	Занимаемая должность
Функции	varchar	MAX	Выполняемые сотрудником функции
Имя компьютера	varchar	MAX	Сетевое имя компьютера
IP	varchar	MAX	IP адрес компьютера
Дата ввода	date	-	Дата ввода в эксплуатацию
Инвентарный	varchar	MAX	Инвентарный номер компьютера
Описание	varchar	MAX	Описание компьютера

В таблице «Комплект» содержатся сведения о комплектующих каждого из компьютеров. Описание полей представлено в таблице 14.

Таблица 14 – Структура таблицы «Комплект»

Название столбца	Тип данных	Длина	Описание
id	int	-	Первичный ключ. Обеспечивает уникальность каждой записи.
id сотрудника	int	-	Внешний ключ. Номер сотрудника (компьютера).
id типа	int	-	Внешний ключ. Номер типа комплектующего.
Модель	varchar	MAX	Модель комплектующего
id производителя	int	-	Внешний ключ. Номер производителя
id поставщика	int	-	Внешний ключ. Номер поставщика.
Стоимость	money	-	Стоимость комплектующего
Гарантия до	int	-	Дата истечения

В таблице «Осмотр» содержатся сведения об осмотрах, комплектующих каждого из компьютеров. Описание полей представлено в таблице 15.

Таблица 15 – Структура таблицы «Осмотр»

Название столбца	Тип данных	Длина	Описание
id	int	-	Первичный ключ. Обеспечивает уникальность каждой записи.
id комплектующего	int	-	Внешний ключ. Номер производителя
Дата осмотра	date	-	Дата осмотра
id мастера	int	-	Номер мастера, производившего осмотр.
Неисправность	varchar	MAX	Выявленная неисправность
Замена	bit	-	Признак необходимости замены
Заказ	bit	-	Признак необходимости заказа комплектующего

В таблице «Типы» содержатся наименования типов (категорий) комплек-

тующих. Описания полей представлено в таблице 16.

Таблица 16 – Структура таблицы «Типы»

Название столбца	Тип данных	Длина	Описание
id	int	-	Первичный ключ. Обеспечивает уникальность каждой записи.
Тип	varchar	MAX	Наименование типа (категории) комплектующего
Характеристика	varchar	MAX	Характеристика типа комплектующего

В таблице «Производители» содержатся наименования о производителях комплектующих. Описания полей представлено в таблице 17.

Таблица 17 – Структура таблицы «Производители»

Название столбца	Тип данных	Длина	Описание
id	int	-	Первичный ключ. Обеспечивает уникальность каждой записи.
Производитель	varchar	MAX	Наименование производителя комплектующего
Страна	varchar	MAX	Страна-производитель комплектующего

В таблице «Проекты» содержатся сведения о проектах, выполняемых организацией. Описания полей представлено в таблице 18.

Таблица 18 – Структура таблицы «Проекты»

Название столбца	Тип данных	Длина	Описание
id	int	-	Первичный ключ. Обеспечивает уникальность каждой записи.
Название проекта	varchar	MAX	Название проекта
id Сотрудника	int	-	Номер заказчика
Описание проекта	varchar		Характеристика выполняемого проекта.
Начало	date	-	Дата начала работ по проекту
Окончание	date	-	Дата окончания работ по проекту
Стоимость	money	-	Стоимость выполнения проекта
Состояние	varchar	-	Состояние работ над проектом

В таблице «Выполнение» содержатся сведения о сотрудниках, задействованных в выполнении проекта. Описания полей представлено в таблице 19.

Таблица 19 – Структура таблицы «Выполнение»

Название столбца	Тип данных	Длина	Описание
id	int	-	Первичный ключ. Обеспечивает уникальность каждой записи.
id сотрудника	int	-	Внешний ключ. Номер сотрудника (компьютера).
id проекта	int	-	Внешний ключ. Номер проекта.
Характеристика	varchar	MAX	Роль сотрудника в проекте

В таблице «Работы» содержатся данные о работе сотрудника по проекту. Описания полей представлено в таблице 20.

Таблица 20 – Структура таблицы «Работы»

Название столбца	Тип данных	Длина	Описание
id	int	-	Первичный ключ. Обеспечивает уникальность каждой записи.
id выполнения	int	-	Внешний ключ. Номер выполняемого сотрудником проекта.
Дата	date	-	Дата выполнения работы
Характеристика	varchar	MAX	Характеристика выполняемой работы

В таблице «Поставщики» содержатся данные о поставщиках и заказчиках (Сотрудниках) предприятия. Описания полей представлено в таблице 21.

Таблица 21 – Структура таблицы «Поставщики»

Название столбца	Тип данных	Длина	Описание
id	int	-	Первичный ключ. Обеспечивает уникальность каждой записи.
Наименование	varchar	MAX	Наименование поставщика/заказчика сотрудником проекта.
ИНН	varchar	MAX	ИНН Сотрудника
КПП	varchar	MAX	КПП Сотрудника
Контактная информация	varchar	MAX	Контактная информация о Сотруднике
Банк	varchar	MAX	Банковские реквизиты Сотрудника
Руководитель	varchar	MAX	Руководитель организации

В таблице «Мастера» содержатся данные о мастерах, производящих осмотр и ремонт компьютеров. Описания полей представлено в таблице 22.

Таблица 22 – Структура таблицы «Мастера»

Название столбца	Тип данных	Длина	Описание
id	int	-	Первичный ключ. Обеспечивает уникальность каждой записи.
ФИО мастера	varchar	MAX	ФИО мастера
Квалификация	varchar	MAX	Квалификация мастера

2.6 Описание структуры программы

Программное обеспечение для автоматизации работы фирмы по обслуживанию компьютеров состоит из 13 модулей. Большинство из них служат для работы со справочной информацией.

Модуль главного меню программы UnMenu выполняет навигационные функции, т.е. посредством этого модуля вызываются все остальные, в которых происходит выполнение основных функций программы.

Модуль для работы с поставщиками и клиентами, предназначены для отображения, добавления, редактирования, поиска и удаления данных о поставщиках и клиентах организации комплектующих с представлением необходимых реквизитов (модули UnClient).

Модуль UnMaster служит для ведения справочника лиц, занимающихся обслуживанием оборудования.

Модуль UnType предназначен для ведения реестра типов комплектующих.

Модуль UnSProject служит для ведения справочника разрабатываемых организацией проектов.

Модуль UnManuf служит для ведения справочника производителей оборудования.

Модуль для учета компьютерной и копировально-множительной техники (UnComp) предназначен для отображения, добавления, редактирования, удаления и поиска данных о комплектующих с указанием основных характеристик.

Модуль для учета разработки проектов (UnProject) и выполняемыми по ним работам предназначены для отображения, добавления, редактирования и удаления данных о проектах, разрабатываемых по договорам с заказчиками, а также выполнении работ по соответствующим проектам с указанием типов выполняемых работ, состояния работ, сотрудников, задействованных на кадрах из проектов.

Модуль UnRate обеспечивает формирование диаграммы исходя из данных отчета, а именно графическое отображение рейтинга сотрудников, исходя

из задействованных в проектах.

Доступ программы к базе данных реализуется с помощью технологии ADO (Microsoft ActiveX Data Objects), которая обеспечивает универсальный доступ к источникам данных из приложений БД. В модуле данных UnDM использованы следующие компоненты данной технологии:

- ADOConnection – обеспечивает доступ к источнику данных и управляет соединением с объектами хранилища данных;
- ADODataSet – предназначен для представления набора данных из хранилища данных ADO;
- ADOTable – обеспечивает использование в приложении таблиц БД;
- ADOQuery – обеспечивает применение запросов SQL при работе с данными через ADO.

Модуль данных UnDM использует данную технологию и содержит невидимые компоненты для работы с базой данных и отвечает за связь приложения с базой данных. Данный модуль используется при работе процедур всех вышеперечисленных модулей программы.

2.7 Описание интерфейса

Разрабатываемая информационная система должна обеспечивать выполнение следующих основных функций:

- учет состояния вычислительной и копировально-множительной техники предприятия;
- прогнозирование оптимального количества запасных частей для восстановления вычислительной и копировально-множительной техники предприятия;
- планирование и контроль выполнения проектов;
- формирование рейтинга специалистов отдела.

Для работы с программой необходимо запустить исполняемый файл программы ASU.EXE.

При запуске программы открывается её главное окно, содержащее меню. С помощью выбора одного из пунктов задаётся режим работы, т.е. иницирует-

ся переход в другие окна, через которые и осуществляется реализация основных функций программы, рисунок 34.



Рисунок 34 – Главное окно программы

При выборе пункта меню «Клиенты/Поставщики» открывается окно, содержащее следующая информация: в левой части окна отображается наименование фирм-поставщиков, а в правой части окна отображаются данные выбранного поставщика. В нижней части окна отображены элементы управления, а именно навигационный интерфейс, элементы для обеспечения поиска данных по критериям, а также кнопка закрытия данной формы.

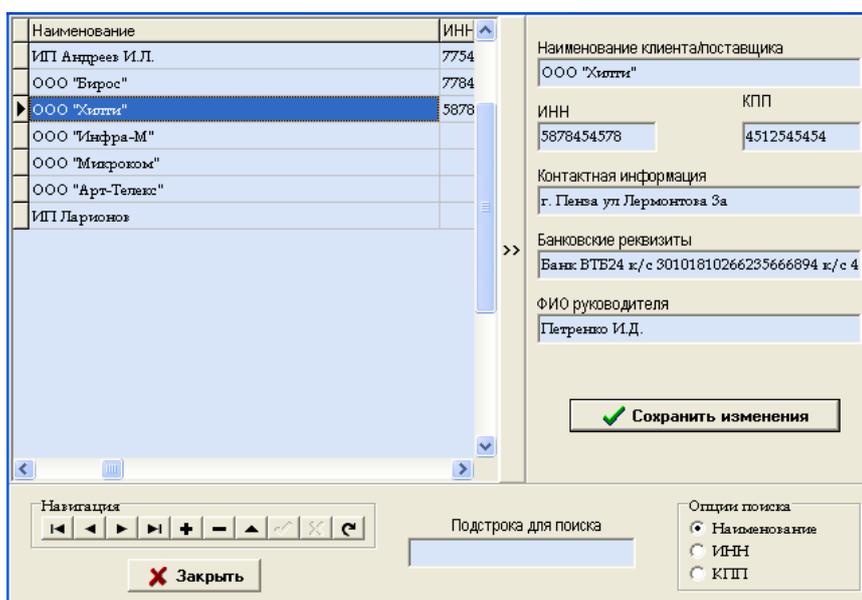


Рисунок 35 – Справочник клиентов/поставщиков

На данной форме необходимо заполнить все необходимые поля и нажать кнопку «Сохранить изменения». При условии, что все поля заполнены, информация о новом поставщике сохраняется в базе данных.

При выборе пункта меню «Производители» откроется окно справочника производителей, рисунок 35. В нижней части окна отображены элементы управления, а именно навигационный интерфейс, элементы для обеспечения поиска данных по заданным критериям, а также кнопка закрытия данной формы.

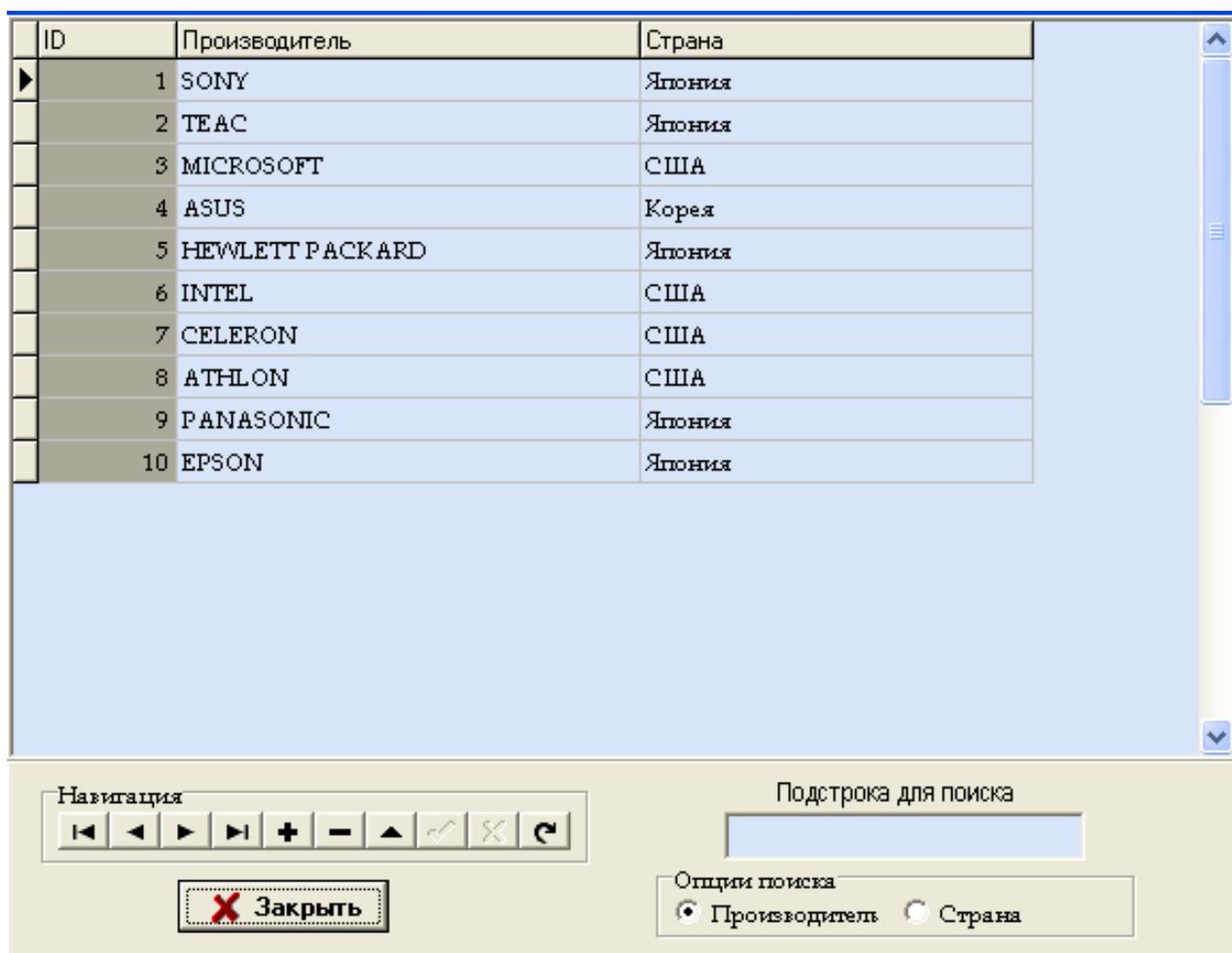


Рисунок 36 – Справочник производителей

При выборе пункта меню «Типы комплектующих» откроется окно справочника типов (категорий) комплектующих, рисунок 36. В нижней части окна отображены элементы управления, а именно навигационный интерфейс, элементы для обеспечения поиска данных по заданным критериям, а также кнопка закрытия формы.

ID	Название типа	Характеристика
1	Материнская плата	Платформа для установки остальных устройств
2	Процессор	Ядро системы
3	Видеокарта	Отвечает за формирование изображения
4	Звуковая карта	Воспроизведение и обработка звуковых эффектов
5	Жесткий диск	Хранение информации
6	Оперативная память	Содержит информация, обработ. в данный момент
7	Монитор	Экран для формирования граф. образов
8	Принтер	Устройство вывода на печать
9	Факс	аппарат для приема текстовой информации
10	Копировальный аппарат	Копирование бумажных документов

Навигация: [←] [▶] [⏪] [⏩] [⏴] [⏵] [↶] [↷] [↺] [↻]

Подстрока для поиска по названию:

X Закрыть

Рисунок 37 – Справочник типов комплектующих

При выборе пункта меню «Мастера» откроется окно справочника мастера, которые осуществляют сервис компьютерной т копировально-множительной техники, рисунок 37. В нижней части окна отображены элементы управления, а именно навигационный интерфейс, элементы для обеспечения поиска данных по заданным критериям, а также кнопка закрытия формы.

ID	ФИО мастера	Квалификация
1	Савикин Дмитрий Николаевич	Старший инженер
2	Иванько Олег Николаевич	Инженер I кат
3	Баусов Юрий Валентинович	Наладчик
4	Некрасов Андрей Викторович	Старший программист

Навигация: [←] [▶] [⏪] [⏩] [⏴] [⏵] [↶] [↷] [↺] [↻]

Подстрока для поиска:

Опции поиска: ФИО мастера Квалификация

X Закрыть

Рисунок 38 – Справочник мастеров

При выборе пункта меню «Проекты» открывается окно, содержащее следующую информацию: в левой части окна отображается название проекта, а в правой части окна отображаются данные по выбранному проекту, рисунок 38. В нижней части окна отображены элементы управления, а именно навигационный интерфейс, элементы для обеспечения поиска данных по заданным критериям, а также кнопка закрытия данной формы.

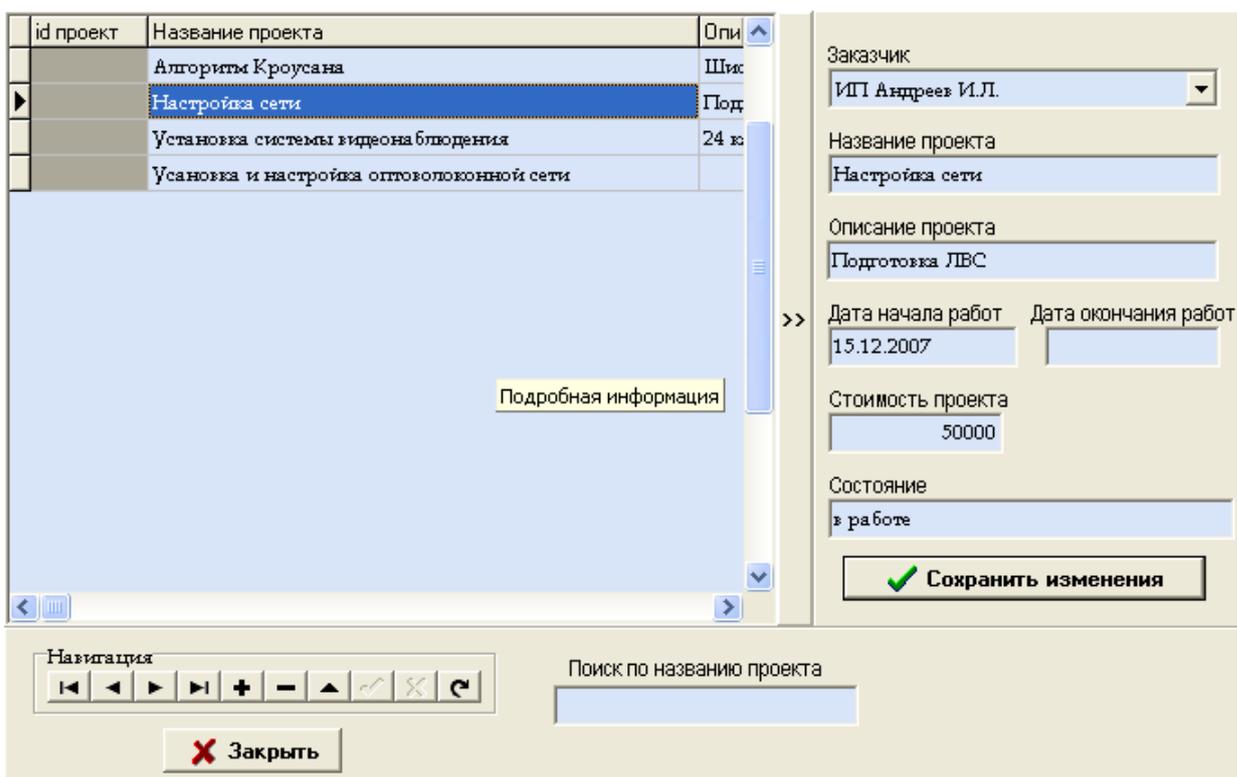


Рисунок 39 – Справочник мастеров

При выборе пункта меню «Учет»-> «Учет компьютерной техники» на экране появится окно, изображенное на рисунке 39. В данном окне осуществляется учет компьютерной техники по отделам и рабочим местам организации. В верхней части формы представлены сведения об отделах организации, а ниже расположена подчиненная таблица, содержащая сведения о сотрудниках отдела и закрепленных за ними компьютерах. При этом панель управления сведениями о сотруднике и его компьютере расположена для удобства в левой части и имеется возможность убрать ее, путём нажатия на кнопку “>>”. В нижней части окна отображены элементы управления, а именно навигационный интерфейс, элементы для обеспечения поиска данных по заданным критериям, а также

кнопка закрытия данной формы.

ID	Название отдела	ФИО начальника
1	Отдел сопровождения	Иванов Сергей Иванович
2	Отдел стандартизации	Конов Олег Иванович
3	Отдел монтажа	Петрушин Олег Владимирович
4	Отдел программирования	Смагин Алексей Олегович

ID	ФИО сотрудника	Должность
1	Иванов Сергей Иванович	нач. отдела

Компьютерная техника в отделе		
ID	ФИО сотрудника	Должность
1	Иванов Сергей Иванович	нач. отдела

Навигация по сотрудникам/компьютерам

Подстрока для поиска

Опции поиска

- ФИО сотрудника
- Имя компьютера
- Должность
- Инвентарный номер

Закреть **Комплектующие компьютера** **Сохранить изменения**

Рисунок 40 – Форма учета компьютерной техники

Для того, чтобы посмотреть список комплектующих и внешних устройств выбранного компьютера необходимо нажать кнопку «Комплектующие компьютера». При этом на экране появится форма учета комплектующих и прочих прочей техники по выбранному сотруднику/компьютеру. В нижней части окна отображены элементы управления, а именно навигационный интерфейс, элементы для обеспечения поиска данных по заданным критериям, а также кнопка закрытия данной формы.

Тип комплектующего	Модель
Видеокарта	ATI REDEON 5600

Тип комплектующего	
Тип комплектующего	Модель
Видеокарта	ATI REDEON 5600
Производитель	ASUS
Поставщик	ООО "Микроком"
Дата выпуска	Гарантия до
22.12.2007	22.12.2008
Стоимость	3400

Навигация

Поиск по модели комплектующего

Закреть **История осмотров и ремонтов**

Рисунок 41 – Форма учета комплектующих по выбранному компьютеру.

Для просмотра/редактирования сведений о сервисе каждого комплектующего/устройства необходимо нажать на кнопку «История осмотров и ремонтов». В результате чего на экране отобразится форма с данными о сервисном обслуживании выбранного устройства. Данная форма представлена на рисунке 41. При этом панель управления сведениями о сервисном обслуживании устройства расположена для удобства в левой части и имеется возможность убрать ее, путём нажатия на кнопку “>>”. Там же можно установить признаки сервисного обслуживания («флажки»), а именно: «Необходимость замены», «По гарантии», «Необходимость заказа». В нижней части окна отображены элементы управления, а именно навигационный интерфейс, элементы для обеспечения поиска данных по заданным критериям, а также кнопка закрытия данной формы.

Дата_осмотра	ФИО мастера
25.12.2007	Некрасов Андрей Викторович

Дата осмотра: 25.12.2007
 ФИО мастера: Некрасов Андрей Викторович
 Вид обслуживания: установка
 Необходимость замены
 По гарантии
 Необходимость заказа

Навигация: [Назад] [Плюс] [Минус] [Вперед] [Сброс] [Обновить]

Опции фильтра:
 Сервис в период с 22.12.2007 по 22.12.2007
 Необходимость замены Необходимость заказа
 По гарантии

Рисунок 42 – История сервисного обслуживания выбранного устройства

При выборе пункта меню «Учет»-> «Учет проектов» на экране появится окно «Учет разработки проектов», изображенное на рисунке 42.

Информация о проекте

Название проекта: Алгоритм Кроусана

Описание проекта: Шифрование данных закрытым ключом

Дата начала работ: 09.12.2007

Сотрудники на проекте/выполняемые функции

Сотрудник	Характеристика
Конов Олег Иванович	Разработка интерфейса
Иванов Сергей Иванович	Разработка модулей связи

Выполняемые по проекту работы

Дата	Характеристика выполняемой работы
10.12.2007	Разработка концепции интерфейса

Рейтинг сотрудников

Закреть

Рисунок 43 – Учет разработки проектов

В данном окне осуществляется учет разработки проектов, роль каждого сотрудника в каждом из проекта, а также работы, выполняемые в рамках каждого проекта. В верхней части формы представлены сведения о проектах, разрабатываемых организацией, а ниже расположена подчиненная таблица, содержащая сведения о сотрудниках, которые задействованы на каждом из проектов. Ниже расположены работы, выполняемые в рамках данного проекта каждым из сотрудников работы. В нижней части окна отображены элементы управления, а именно навигационный интерфейс, а также кнопка закрытия данной формы. Для просмотра рейтинга сотрудников следует нажать кнопку «Рейтинг сотрудников», при этом на экране появится одноименная форма, в которой в виде столбовой диаграммы изображен рейтинг сотрудников, исходя из участия их в каждом из проектов. Внешний вид диаграммы представлен на рисунке 43.

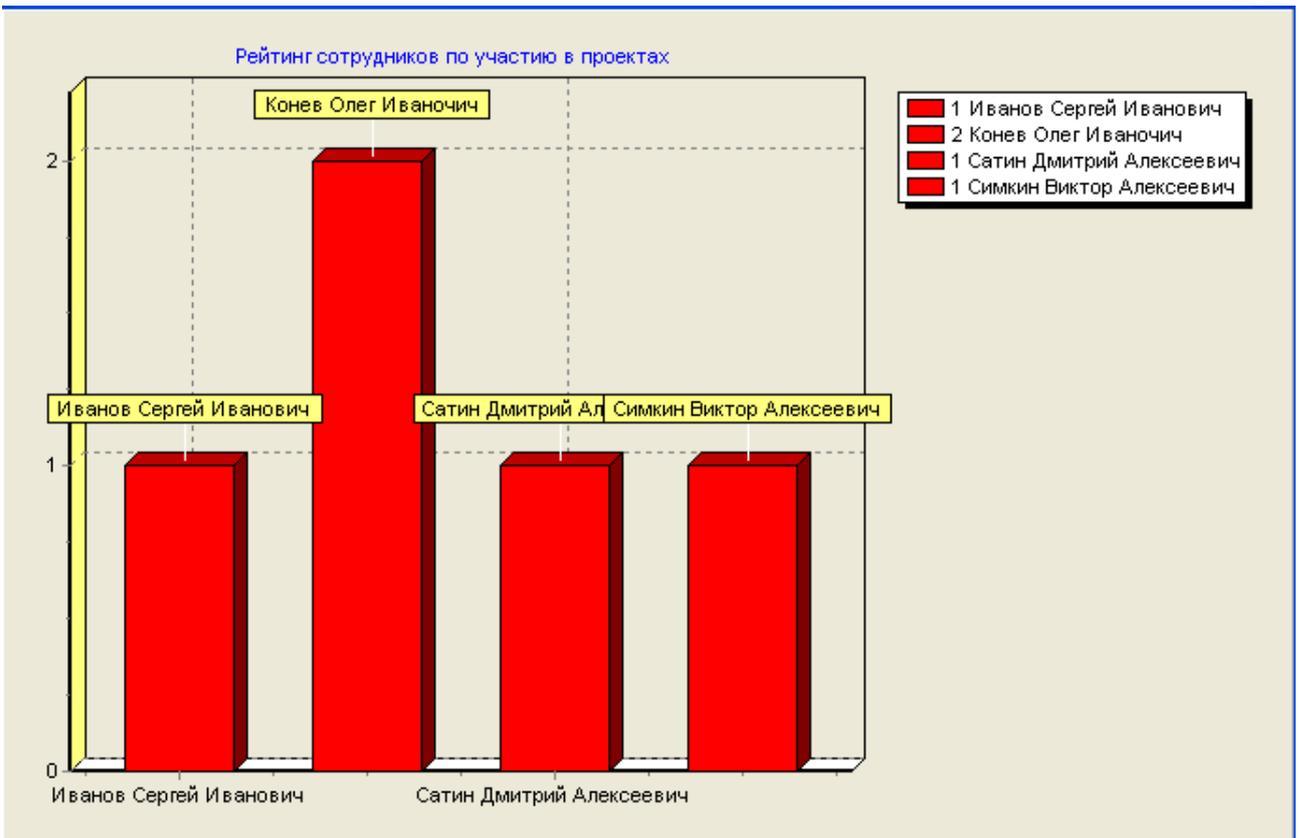


Рисунок 44 – Диаграмма рейтинга сотрудников

Для формирования отчета, содержащего сведения о заказчиках и заказываемых ими проектах необходимо выбрать пункт меню «Отчеты»-> «Проекты по заказчикам». При этом на экране появится отчет в формате MS Excel следующего вида.

1	07.06.2017						
2	Название заказчика	Название проекта	Описание проекта	Стоимость	Дата начала	Дата окончания	Состояние проекта
3	ИП Андреев И.Л.	Настройка сети	Подготовка ЛВС	50000	15.12.2007		в работе
4	ООО "Бирос"	Алгоритм Кроусана	Шифрование данных закрытым ключом	800000	09.12.2007		
5	ООО "Микроком"	Усановка и настройка оптоволоконной сети		0	17.12.2007		
6	ООО "Арт-Телекс"	Установка системы видеонаблюдения	24 камеры по периметру	0	20.12.2007		работы начаты

Рисунок 45 – Отчет «Проекты по заказчикам»

Для формирования отчета, содержащего сведения о сотрудниках, задействованных в каждом из проектов необходимо выбрать пункт меню «Отчеты»-

> «Сотрудники, задействованные в проектах». При этом на экране появится отчет в формате MS Excel следующего вида.

07.06.2017			
Название проекта	ФИО сотрудников, задействованных на проекте	Выполняемые сотрудниками функции	
Алгоритм Кроусана	Конев Олег Иванович	Разработка интерфейса	
	Иванов Сергей Иванович	Разработка модулей связи	
Настройка сети	Конев Олег Иванович	Методические рекомендации	
	Симкин Виктор Алексеевич	Отладка системы	
Установка системы видеонаблюдения			
	Сатин Дмитрий Алексеевич	Обследование объекта	
Усановка и настройка оптоволоконной сети			

Рисунок 46 – Отчет «Сотрудники, задействованные в проектах»

Для формирования отчета, содержащего сведения о сервисном обслуживании компьютеров необходимо выбрать пункт меню «Отчеты»-> «Сервис компьютеров». При этом на экране появится отчет в формате MS Excel следующего вида.

07.06.2017						
Сотрудник/имя компьютера	Модель/тип комплектующего	Дата осмотра	ФИО мастера	Вид сервиса	Примечание	
Иванов Сергей Иванович/	Core2Duo 2000/Процессор	08.07.2007	Иванько Олег Никол	Установка	гарантия заказ	
Конев Олег Иванович/StandKon	ATI REDEON 5600/Видеокарта	25.12.2007	Некрасов Андрей Ви	установка	замена гарантия	
Симкин Виктор Алексеевич/StandSim						
Петрушин Олег Владимирович/MontPetr						
Сатин Дмитрий Алексеевич/MontSat						
Смагин Алексей Олегович/ProgSma						
Ивашкин Олег Федорович/Proglv	ACORP 4566/Материнская плата					
	Intel Core2DUO 2000/Процессор					
Самсонов Павел Викторович/ProgSam						

Рисунок 47 – Отчет «Сервис компьютеров»

2.8 Входные и выходные данные

Входными данными должны быть:

- по разрабатываемым проектам – наименование клиента, название проекта, описание проекта, дата начала работ, дата окончания работ, стоимость разрабатываемого проекта;
- данные о компьютерной и копировально-множительной техники (расходных материалах, запчастях) – наименование, производитель, поставщик, ответственное лицо, сведения о сервисе и ремонте.

Выходными данными должны быть:

- сотрудники, задействованные в каждом из проектов;
- история сервиса каждого компьютера;
- список всех комплектующих, находящихся на гарантии;
- комплектующие, которые необходимо заказать;
- комплектующие, которые необходимо заменить;
- список проектов по заказчикам;
- рейтинг сотрудников.

3 ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА

3.1 Прямая и косвенная экономическая эффективность

Эффективность – это одно из наиболее важных экономических понятий. Эффективность – одна из возможных характеристик качества системы, а именно ее характеристика с точки зрения соотношения затрат и результатов функционирования системы.

Возникает необходимость количественной оценки положительного влияния на хозяйственный процесс автоматизации задач планирования, учета и анализа, и других расчетов. Сопоставление этого эффекта с производственными затратами позволяет определить экономическую эффективность.

Чтобы выявить и рассчитать экономическую эффективность, надо знать, в каких показателях могут быть выражены результаты внедрения информационной системы. Оценка ее состоит в определении ряда показателей, характеризующих использование различных видов ресурсов, повышение качества и снижение себестоимости обработки, роста производительности труда. Одни из показателей дают оценку прямого эффекта от применения информационных систем, другие косвенно характеризуют экономическую эффективность. Общая экономическая эффективность складывается из прямой и косвенной эффективности.

Прямая эффективность при автоматизации учета связана с сокращением затрат труда и различных расходов непосредственно на обработку данных. Показатели ее могут быть измерены и выражены в количественных величинах.

Косвенная эффективность проявляется в улучшении работы персонала благодаря использованию всесторонней и более качественной информации, и отражается на показателях хозяйственной деятельности предприятия. Количественно измерить влияние автоматизированной обработки данных на результаты хозяйственной деятельности не всегда возможно, так как повышение эффективности зависит не только от применения технических средств, но и от влияния других многочисленных факторов.

Обобщающими показателями прямой эффективности, являются:

- срок окупаемости единовременных затрат;
- коэффициент эффективности.

Показатели косвенной эффективности - это показатели, которые могут отражать качественные изменения при автоматизации управленческих работ:

- повышение качества и производительности труда;
- снижение трудоемкости работ.

Таким образом, из перечисленных выше показателей эффективности к основным показателям, которые используются для оценки экономического эффекта от внедрения информационной системы, относятся:

- экономический эффект;
- период окупаемости.

Экономический эффект – результат внедрения, какого-либо мероприятия, выраженный в стоимостной форме, в виде экономии от его осуществления.

Период окупаемости (величина, обратная коэффициенту эффективности) – показатель эффективности капитальных вложений - представляет собой период времени, в течение которого производственные затраты на ИС окупаются полученным эффектом.

Так, для предприятия ООО «Эрлант», при использовании разработанной системы, основными источниками экономии станут:

- уменьшение затрат в отличие от приобретения аналога;
- увеличение объемов обработки документов за счет автоматизации обработки;
- снижение трудоемкости работы сотрудников с последующим уменьшением ошибок.

Также можно выделить социально-значимый характер от применения новой системы, который заключается в том, что данная система, посредством структурированности данных, позволяет оперативно, в автоматическом режиме, находить и обрабатывать необходимые документы. Увеличив, тем самым, объемы обработки документов, уменьшив количество ошибок и снизив трудо-

емкость работ.

3.2 Определение затрат на разработку и внедрение программного продукта

Полные затраты на выполнение разработки складываются из следующих компонентов:

$$Z = Z_M + Z_{ZO} + Z_{ЗД} + Z_C + Z_{МВ} + Z_H \quad (1)$$

где Z_M – стоимость материалов, покупных изделий;

Z_{ZO} – основная заработная плата;

$Z_{ЗД}$ – дополнительная заработная плата;

Z_C – отчисления на социальные нужды;

$Z_{МВ}$ – затраты на машинное время;

Z_H – накладные расходы.

Определим конкретное числовое значение каждого из этих параметров.

3.2.1 Стоимость материалов

В данную статью затрат мы включаем покупку программного обеспечения.

В отношении статей затрат предлагаемой разработки нужно выделить следующее: в качестве базовой платформы для программирования выбрана Microsoft Borland Delphi 7, права на лицензионное использование которой уже имеются. Это означает, что из статей затрат исключается статья на приобретение лицензионного программного обеспечения.

3.2.2 Основная заработная плата

Труд исполнителя проекта оплачивается исходя из средней месячной заработной платы программиста, не имеющего опыта работы и проживающего в Амурской области. По данным сайта www.gks.ru она составляет 36000 рублей.

На разработку данного программного продукта было потрачено 28 дней. Посчитаем это же время в месяцах, учитывая, что в месяце в среднем 4,5

недели, а в неделе 5 рабочих дней:

$$B = \frac{167}{4,5 \cdot 5} \approx 1,2$$

Разработкой программного продукта занимался один программист. Таким образом, основная заработная плата определяется следующим образом:

$$Z_{zo} = 30000 \times 1,2 = 360000 \text{ (руб.)}$$

3.2.3 Накладные расходы

Накладные расходы – это расходы по управлению и обслуживанию производства: затраты на содержание и эксплуатацию основных средств, на управление, организацию, обслуживание производства, непроизводительные расходы (потери от простоев, порчи материальных ресурсов и т.п.). Накладные расходы включаются в себестоимость продукции, издержки ее производства и обращения.

Накладные расходы носят комплексный характер, то есть включают различные экономические элементы затрат. Учет накладных расходов ведется на счетах общепроизводственных расходов и общехозяйственных расходов, которые являются собирательно-распределительными и закрываются по окончании отчетного периода. По каждому виду расходов составляется смета с подразделением по статьям.

В нашем случае накладные расходы составляют 40 % от полной заработной платы и включают в себя: транспортные расходы, зарплату руководителю проекта, организацию внутренней информационной сети, обучение обслуживающего персонала и другие расходы.

Сведем итоговую смету затрат в таблицу 23.

Таблица 23 – Итоговая смета затрат

Наименование статьи расходов	Сумма в рублях
Стоимость материалов	500
Основная заработная плата	36000
Отчисления на социальные нужды	10800
Стоимость машинного времени	1344
Накладные расходы	20592
Итого	69236

3.3 Выбор метода расчета

Количественную оценку эффективности программного продукта целесообразно производить путем сравнения затрат текущего (базового) осуществления работ с внедряемым вариантом путем определения параметров обоих вариантов.

Годовой экономический эффект определяется по формуле:

$$\mathcal{E}_2 = T_{\bar{6}} - T_{\bar{e}} , \quad (2)$$

где $T_{\bar{6}}$ – годовые текущие затраты в базовом варианте;

$T_{\bar{e}}$ – годовые текущие затраты во внедряемом варианте.

Срок окупаемости затрат в годах определяется по формуле:

$$T_{ок} = \frac{K_{\bar{e}}}{\mathcal{E}_2} , \quad (3)$$

где $K_{\bar{e}}$ – капитальные вложения во внедряемом варианте.

3.3.1 Расчет текущих затрат

Текущие расходы складываются из заработной платы работников и других расходов, связанных с выполнением проводимых работ. Заработная плата определяется как сумма основной и дополнительной заработной платы, а также отчислений на социальное страхование и в пенсионный фонд.

В базовом варианте заняты два человека с основной зарплатой в 72000 рублей.

Рассчитаем годовую основную заработную плату на всех двух сотрудников:

$$Z_o = 36000 \times 12 \times 2 = 864000 \text{ (руб.)}$$

Дополнительная заработная плата на двух сотрудников, выполняющих ручные расчеты:

$$Z_o = Z_o \times 0,1 = 864000 \times 0,1 = 86400 \text{ (руб.)}$$

Отчисления на социальные нужды посчитаем следующим образом:

$$Z_c = (Z_o + Z_d) \times 0,3 = 285120 \text{ (руб.)}$$

Просуммируем полученные значения:

$$Z_e = 864000 + 285120 = 1235520 \text{ (руб.)}$$

Накладные расходы в базовом варианте:

$$H_b = 1235520 \times 0,4 = 494208 \text{ (руб.)}$$

Следовательно, текущие расходы в базовом варианте составляют:

$$T_b = Z_b + H_b = 1729728 \text{ (руб.)}$$

Рассмотрим внедряемый вариант. Во внедряемом варианте занят только один человек с зарплатой в 36000 рублей, поскольку время, затрачиваемое на подготовку аналогичной документации, сокращается в несколько раз. Рассчитаем основную и дополнительную годовую заработную плату специалистов, а также отчисления на социальные нужды.

Основная заработная плата:

$$Z_o = 36000 \times 12 \times 2 = 432000 \text{ (руб.)}$$

Дополнительная заработная плата:

$$Z_d = Z_o \times 0,1 = 432000 \times 0,1 = 43200 \text{ (руб.)}$$

Отчисления на социальные нужды:

$$Z_c = (Z_o + Z_d) \times 0,3 = 142560 \text{ (руб.)}$$

Рассчитаем затраты во внедряемом варианте:

$$Z_e = 432000 + 43200 + 142560 = 617760 \text{ (руб.)}$$

Накладные расходы:

$$H_v = 617760 \times 0,4 = 247104 \text{ (руб.)}$$

Таким образом, расходы во внедряемом варианте составляют:

$$T_v = Z_v + H_v = 864864 \text{ (руб.)}$$

3.3.2 Расчет экономического эффекта

Годовой экономический эффект от использования программного продукта составит:

$$Э_2 = T_b - T_e = 864864 \text{ (руб.)}$$

Экономический эффект от внедрения программного продукта очевиден,

он позволяет сэкономить более 800000 рублей в год. Это возможно благодаря сокращению числа специалистов, занимающихся расчетами, уменьшению заработной платы, поскольку специалистам потребуется прилагать меньше усилий благодаря автоматизации и повышению скорости выполнения работ.

Значение годового экономического эффекта позволяет сделать вывод о том, что внедрение системы экономически целесообразно. Кроме того, при принятии решения о целесообразности внедряемой системы, необходимо учитывать также достигаемое повышение технического уровня производства, повышение качества и достоверности выдаваемой информации, а также социальный эффект, состоящий в замене ручного труда более высококвалифицированным.

ЗАКЛЮЧЕНИЕ

В ходе выполнения проектирования был проведен анализ деятельности ООО «Эрлант» с представлением общей характеристики, организационной структуры и структуры управления предприятия, анализа документооборота и характеристики объекта автоматизации. Изучены принципы ведения учета проектных работ, количественного учета поступления и расхода комплектующих и расходных материалов.

Разработана функциональная модель АСУ отдела информационных технологий с использованием CASE - средства BPWin, проведено инфологическое проектирование, разработана структура базы данных с использованием ERWin.

Реализация системы проведена с использованием СУБД Firebird SQL, интерфейс приложения разработан в среде Borland Delphi 7.0.

На этапе проектирования автоматизированной системы была разработана структура системы, дана характеристика функциональных и обеспечивающих систем, также была разработана информационная система.

Результатами разработки являются:

- пользовательский интерфейс АСУ, включающий экранные формы для работы со справочными, оперативными данными и отчетными данными;
- формы документов и отчетной документации для автоматизированной подготовки на рабочем месте требуемой информации;
- запросы, обеспечивающие формирование данных для первичных и отчетных документов;
- руководство пользователя.

После этапа разработки автоматизированной системы была проведена оценка экономической эффективности системы, которая показала, что применение данной системы положительно сказывается на работе предприятия. Значение годового экономического эффекта и восьмимесячный срок окупаемости затрат позволяет сделать вывод о том, что внедрение системы экономически цели-сообразно.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Буч, Г., Максимчук, Р., Энгл М. Объектно-ориентированный анализ и проектирование с примерами приложений. – М.: Вильямс, 2012. – 720 с.
- 2 Винсенс, Т. Firebird. Библиотека профессионала / Т. Винсенс. — М.: Символ-плюс, 2010. – 267 с.
- 3 Гнеденко, Б.В. Надежность программного изделия / Б.В. Гнеденко, И.Н. Коваленко. – М.: Мир, 2009. – 320 с.
- 4 Горелик, О.М. Техничко-экономический анализ и его инструментальные средства. Учебное пособие / О.М. Горелик. – М.: Финансы и статистика, 2016. – 240 с.
- 5 Гутманс, Э. Я. РНР 5. Профессиональное программирование / Э. Я. Гутманс. – М.: Символ-плюс., 2012. – 517 с.
- 6 Дейт, К. Введение в системы баз данных: пер. с англ. / К. Дейт.– Киев: Вильямс, 2013. – 1328 с.
- 7 Дейт, К. Дж. Введение в системы баз данных / К.Дж. Дейт. – Киев: Вильямс, 2008. – 846с.
- 8 Димов, Э.М. Проектирование информационных систем: Учеб. пособие / Э.М. Димов, А.Р. Диязитдинова. – Самара: Издательство Поволжского гос. Академии, 2011. – 112 с.
- 9 Инструментальная среда ERWin. [Электронный ресурс]. – Режим доступа: <http://ali-ce.stup.ac.ru/case/caseinfo/erwin/part1.html>. – 16.03.2017.
- 10 Карпова, Т.С. Базы данных: модели, разработка, реализация / Т.С. Карпова. – СПб.: Питер, 2003. – 304 с.
- 11 Коннолли, Т. Базы данных / Т. Коннолли, К. Бегг, А. Страчан. – Киев: Вильямс, 2014. – 111с.
- 12 Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. / Т. Коннолли. – М.: Издательский дом «Вильямс», 2010. – 1120 с.
- 13 Кулаков, Ю.А. Компьютерные сети. Выбор, установка, использование

и администрирование. / Ю.А. Кулаков, С.В. Омелянский – К.: Юниор, 2011. – 544 с.

14 Кумскова, И.А. Базы данных / И.А. Кумскова. – Издательство: Кнорус, 2010. – 496 с.

15 Олифер, В.Г. Основы сетей передачи данных / В.Г. Олифер, Н.А. Олифер.– СПб: Питер, 2009.- 663с.

16 Советов, Б.Я. Базы данных: теория и практика: Учебник для бакалавров / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской. - М.: Юрайт, 2013г., 463 с

17 Фараонов, В.В. Delphi 7. Учебный курс. / В.В. Фараонов М.: «Нолидж», 2009. – 415 с.

18 Хомоненко, А.Д. Базы данных / А.Д. Хомоненко, В.М. Цыганков, М.Г. Мальцев; под ред. А.Д. Хомоненко. – 6-е изд., доп. – СПб.: КОРОНА, 2009. – 736 с

19 Хомоненко, А.Д. Базы данных / А.Д. Хомоненко. – СПб.: Корона принт, 2007. – 414с.

20 Швецов, В.И. Базы данных / Швецов В.И. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУ-ИТ), 2016. – 218 с.

ПРИЛОЖЕНИЕ А
Техническое задание

1. ОБЩИЕ СВЕДЕНИЯ

1.1 Наименование системы

Полное наименование: Проектирование автоматизированной информационно-управляющей системы ИТ - отдела ООО «Эрлант».

Краткое наименование: АИС.

1.2 Основание для проведения работ

– ГОСТ 34.602-89 – техническое задание на проектирование автоматизированной системы управления;

– требование к системе;

– первичные документы.

1.3 Наименование организации – Заказчика и Разработчика

Заказчик: ООО «Эрлант».

Адрес юридический: ул. Первомайская 3

Разработчик: студент факультета математики и информатики ФГБОУ ВО «Амурского Государственного Университета» Ковальский Георгий Александрович.

Адрес фактический: ул. Островского 38

Телефон: 89248431881

1.4 Плановые сроки начала и окончания работы

Начало работ: 24.03.2017

Срок окончания работ: 15.06.2017

2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

2.1 Назначение системы

Система предназначена для автоматизации ИТ Отдела.

2.2 Цели создания системы

Продолжение ПРИЛОЖЕНИЕ А

Техническое задание

Разрабатываемую автоматизированную систему предполагается использовать на рабочих станциях сервисного центра, используемых с целью обслуживания клиентов, а также работы по составлению рабочего расписания.

3. ХАРАКТЕРИСТИКА ОБЪЕКТОВ АВТОМАТИЗАЦИИ

Объектом автоматизации является консалтинговая компания ООО «Эрлант»

Требования к системе

3.1 Функции системы

Проектируемая система будет выполнять следующие функции:

- данные в системе должны быть организованы в виде таблиц с реляционной структурой связи;
- компоненты системы должны обмениваются при помощи общей среды передачи данных и глобальных для них переменных, отражающих свойства системы;
- подсистема должна быть разработана с учетом возможной интеграции со смежными системами, что подразумевает ее разработку в соответствии с общепринятыми нотациями и форматами конвертации и преобразования данных;
- в системе должны быть использованы по возможности общесоюзные и зарегистрированные республиканские, отраслевые классификаторы, унифицированные документы и классификаторы, действующие на данном предприятии;
- к структуре процесса сбора, обработки, передачи данных в системе и представлению данных предъявляются требования интерфейсной понятности и удобства, а также корректности их обработки.

3.2 Стандарты

Разработка системы регламентируется стандартами:

- ГОСТ 19.001-77 – Общие положения;

Продолжение ПРИЛОЖЕНИЕ А

- ГОСТ 19.004-80 – Термины и определения;
- ГОСТ 19.101-77 – Виды программ и программных документов;
- ГОСТ 19.102-77 – Стадии разработки;
- ГОСТ 19.103-77 – Обозначение программ и программных документов;
- ГОСТ 19.104-78 – Основные надписи;
- ГОСТ 19.105-78 – Общие требования к программным документам;
- ГОСТ 19.106-78 – Требования к программным документам, выполненным печатным способом;
- ГОСТ 19.402-78 – Описание программы;
- ГОСТ 19.502-78 – Описание применения. Требования к содержанию и оформлению;
- ГОСТ 19.505-79 – Руководство оператора. Требования к содержанию и оформлению;
- ГОСТ 19.508-79 – Руководство по техническому обслуживанию. Требования к содержанию и оформлению;
- ГОСТ 34.602-89 (Техническое задание на создание автоматизированной системы);
- ГОСТ 34.201-89 (Виды, комплектность и обозначение документов при создании автоматизированных систем);
- ГОСТ 24.104-85 (Автоматизированные системы управления. Общие требования);
- ГОСТ 34.601-90 (Автоматизированные системы. Стадии создания);
- ГОСТ 25.861-83 (АСУ. Требования по безопасности средств вычислительной техники).

3.3 Структура системы

В системе применяется язык программирования высокого уровня Delphi v7.0. NET Framework определяет среду, которая поддерживает развитие и вы-

Продолжение ПРИЛОЖЕНИЯ А

полнение платформ независимых приложений. Она допускает совместную работу в приложении различных языков программирования, а также обеспечивает переносимость программ и общую модель программирования на Windows. .NET Framework не ограничена платформой Windows, и написанные для этой платформы программы могут быть в дальнейшем перенесены на другие платформы. Delphi v7.0 .NET создает приложения с малым размером скомпилированного кода, имеет встроенный оптимизатор программ, отладчик, большое количество компонентов для визуальной разработки, возможность вставки кода, написанного на машинном языке программирования.

3.4 Требования к техническому обеспечению

Техническое обеспечение серверной части должно удовлетворять следующим требованиям:

- процессор Intel core i3 2GHz и новее;
- ОЗУ 3 гб и выше;
- размер массива данных от 300 гб.

3.5 Требования к программному обеспечению

Программное обеспечение серверной части должно удовлетворять следующим требованиям:

- наличие операционной системы windows 7 и новее;
- наличие сервера БД Microsoft SQL.

4 СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ

Этапы, которые необходимо выполнить при создании информационной системы:

- исследование предметной области, выделение объекта автоматизации, анализ деятельности предприятия;
- составление технического задания: выявление пожеланий заказчика к разрабатываемой системе, определение технических и программных средств

Продолжение ПРИЛОЖЕНИЯ А

для реализации проекта;

- разработка информационной системы: На этом этапе разрабатывается следующее: физическое, логическое и инфологическое проектирование системы;

- реализация информационной системы.

5 ТРЕБОВАНИЯ К СОСТАВУ И СОДЕРЖАНИЮ РАБОТ ПО ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ К ВВОДУ В ДЕЙСТВИЕ

Перед тем как ввести в эксплуатацию готовую информационную систему разработчик обязан договориться с руководством организации о времени, в течение которого он обязан внедрить разработанную систему. Под внедрением системы понимается совокупность мероприятий, включающих в себя обучение персонала, настройку системы для дальнейшего использования, информирование о порядке проведения работ по сопровождению системы и предоставление им необходимой документации на систему, ознакомление администратора с его обязанностями. Также разработчик обязан предоставить демонстрационную версию программы.

ПРИЛОЖЕНИЕ Б

Фрагменты исходного кода программы

```
program;  
  
uses  
  Forms,  
  UnMenu in 'UnMenu.pas' {FmMenu},  
  UnDM in 'UnDM.pas' {DM: TDataModule},  
  UnClient in 'UnClient.pas' {FmClient},  
  UnManuf in 'UnManuf.pas' {FmManuf},  
  UnType in 'UnType.pas' {FmType},  
  UnMaster in 'UnMaster.pas' {FmMaster},  
  UnComp in 'UnComp.pas' {FmComp},  
  UnSet in 'UnSet.pas' {FmSet},  
  UnHistory in 'UnHistory.pas' {FmHistory},  
  UnFiltHist in 'UnFiltHist.pas' {FmFiltHist},  
  UnProject in 'UnProject.pas' {FmProject},  
  UnSProject in 'UnSProject.pas' {FmSProject},  
  UnRate in 'UnRate.pas' {FmRate};  
  
{ $R *.res }  
  
begin  
  Application.Initialize;  
  Application.CreateForm(TFmMenu, FmMenu);  
  Application.CreateForm(TDM, DM);  
  Application.CreateForm(TFmClient, FmClient);  
  Application.CreateForm(TFmManuf, FmManuf);  
  Application.CreateForm(TFmType, FmType);  
  Application.CreateForm(TFmMaster, FmMaster);  
  Application.CreateForm(TFmComp, FmComp);  
  Application.CreateForm(TFmSet, FmSet);  
  Application.CreateForm(TFmHistory, FmHistory);  
  Application.CreateForm(TFmFiltHist, FmFiltHist);  
  Application.CreateForm(TFmProject, FmProject);  
  Application.CreateForm(TFmSProject, FmSProject);  
  Application.CreateForm(TFmRate, FmRate);  
  Application.Run;  
end.  
  
unit UnClient;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, ExtCtrls, DBCtrls, StdCtrls, Buttons, Grids, DBGrids, Mask, DB;  
  
type  
  TFmClient = class(TForm)  
    Panel1: TPanel;  
    Panel2: TPanel;  
    Panel3: TPanel;  
    Panel4: TPanel;  
    GroupBox1: TGroupBox;  
    DBNavigator1: TDBNavigator;  
    BitBtn1: TBitBtn;  
    DBGrid1: TDBGrid;  
    DBEdit1: TDBEdit;
```

Продолжение ПРИЛОЖЕНИЕ Б
Фрагменты исходного кода программы

```
Label1: TLabel;
  DBEdit2: TDBEdit;
  Label2: TLabel;
  DBEdit3: TDBEdit;
  Label3: TLabel;
  DBEdit4: TDBEdit;
  Label4: TLabel;
  DBEdit5: TDBEdit;
  Label5: TLabel;
  DBEdit6: TDBEdit;
  Label6: TLabel;
  BitBtn2: TBitBtn;
  RadioGroup1: TRadioGroup;
  Label7: TLabel;
  EditFind: TEdit;
  procedure Panel4Click(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
  procedure EditFindChange(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FmClient: TFmClient;

implementation
uses UnDM;
{$R *.dfm}

procedure TFmClient.Panel4Click(Sender: TObject);
begin
  if Panel2.Visible = true then
  begin
    Panel2.Visible := false;
    Panel4.Caption := '<<';
    Panel4.Hint := 'Показать подробную информацию';
  end
  else
  begin
    Panel2.Visible := true;
    Panel4.Caption := '>>';
    Panel4.Hint := 'Скрыть подробную информацию';
  end;
end;

procedure TFmClient.BitBtn1Click(Sender: TObject);
begin
  Close;
end;

procedure TFmClient.EditFindChange(Sender: TObject);
begin
  Case RadioGroup1.ItemIndex of
    0 : DM.TbClient.Locate('Наименование', EditFind.Text, [locaseinsensitive, lopartialkey]);
    1 : DM.TbClient.Locate('ИНН', EditFind.Text, [locaseinsensitive, lopartialkey]);
```

Продолжение ПРИЛОЖЕНИЕ Б

Фрагменты исходного кода программы

```
2 : DM.TbClient.Locate('КПП', EditFind.Text, [locaseinsensitive, lopartialkey]);  
end;  
end;
```

```
procedure TFmClient.BitBtn2Click(Sender: TObject);  
begin  
  If DM.TbClient.Modified then DM.TbClient.Post;  
end;  
  
end.
```

```
unit UnComp;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Grids, DBGrids, Mask, DBCtrls, StdCtrls, ExtCtrls, Buttons, DB,  
jpeg;
```

```
type
```

```
TFmComp = class(TForm)  
  Panel1: TPanel;  
  Label7: TLabel;  
  GroupBox1: TGroupBox;  
  DBNavigator1: TDBNavigator;  
  BitBtn1: TBitBtn;  
  RadioGroup1: TRadioGroup;  
  EditFind: TEdit;  
  Panel2: TPanel;  
  BitBtn2: TBitBtn;  
  Panel3: TPanel;  
  Panel4: TPanel;  
  DBGrid1: TDBGrid;  
  Panel5: TPanel;  
  Panel6: TPanel;  
  GroupBox2: TGroupBox;  
  DBNavigator2: TDBNavigator;  
  Panel7: TPanel;  
  DBGrid2: TDBGrid;  
  GroupBox3: TGroupBox;  
  DBEdit3: TDBEdit;  
  Label3: TLabel;  
  DBEdit2: TDBEdit;  
  Label2: TLabel;  
  DBEdit1: TDBEdit;  
  Label1: TLabel;  
  GroupBox4: TGroupBox;  
  DBEdit4: TDBEdit;  
  Label4: TLabel;  
  Label5: TLabel;  
  DBEdit5: TDBEdit;  
  DBEdit6: TDBEdit;  
  Label6: TLabel;  
  DBEdit7: TDBEdit;  
  Label8: TLabel;  
  DBEdit8: TDBEdit;
```

Продолжение ПРИЛОЖЕНИЕ Б

Фрагменты исходного кода программы

```
Label9: TLabel;
BitBtn3: TBitBtn;
Image1: TImage;
procedure Panel4Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure EditFindChange(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FmComp: TFmComp;

implementation
uses UnDM, UnSet;
{$R *.dfm}

procedure TFmComp.Panel4Click(Sender: TObject);
begin
  if Panel2.Visible = true then
  begin
    Panel2.Visible := false;
    Panel4.Caption := '<<';
    Panel4.Hint := 'Показать подробную информацию';
  end
else
  begin
    Panel2.Visible := true;
    Panel4.Caption := '>>';
    Panel4.Hint := 'Скрыть подробную информацию';
  end;
end;

procedure TFmComp.BitBtn1Click(Sender: TObject);
begin
  close;
end;

procedure TFmComp.BitBtn2Click(Sender: TObject);
begin
  If DM.TbComp.Modified then DM.TbComp.Post;
end;

procedure TFmComp.EditFindChange(Sender: TObject);
begin
  Case RadioGroup1.ItemIndex of
    0 : DM.TbComp.Locate('ФИО сотрудника', EditFind.Text, [locaseinsensitive, lopartialkey]);
    1 : DM.TbComp.Locate('Должность', EditFind.Text, [locaseinsensitive, lopartialkey]);
    2 : DM.TbComp.Locate('Имя компьютера', EditFind.Text, [locaseinsensitive, lopartialkey]);
    3 : DM.TbComp.Locate('Инвентарный номер', EditFind.Text, [locaseinsensitive, lopartialkey]);
  end;
end;

procedure TFmComp.BitBtn3Click(Sender: TObject);
```

Продолжение ПРИЛОЖЕНИЕ Б

Фрагменты исходного кода программы

```
begin
    FmSet.Caption := 'Комплекующие компьютера ' + QuotedStr(DM.TbComp.FieldName('Имя
компьютера')).AsString) + ' сотрудника ' + QuotedStr(DM.TbComp.FieldName('ФИО
сотрудника')).AsString);
    FmSet.ShowModal;
end;

end.

unit UnDM;

interface

uses
    SysUtils, Classes, DB, ADODB;

type
    TDM = class(TDataModule)
        ADOC: TADOConnection;
        TbClient: TADOTable;
        DSClient: TDataSource;
        TbClientid: TAutoIncField;
        TbClientDSDesigner: TWideStringField;
        TbClientDSDesigner2: TWideStringField;
        TbClientDSDesigner3: TWideStringField;
        TbClientDSDesigner4: TWideStringField;
        TbClientDSDesigner5: TWideStringField;
        TbClientDSDesigner6: TWideStringField;
        TbManuf: TADOTable;
        TbManufid: TAutoIncField;
        TbManufDSDesigner: TWideStringField;
        TbManufDSDesigner2: TWideStringField;
        DSManuf: TDataSource;
        DSType: TDataSource;
        TbType: TADOTable;
        TbTypeid: TAutoIncField;
        TbTypeDSDesigner: TWideStringField;
        TbTypeDSDesigner2: TWideStringField;
        TbSet: TADOTable;
        TbMaster: TADOTable;
        TbMasterid: TAutoIncField;
        TbMasterDSDesigner: TWideStringField;
        TbMasterDSDesigner2: TWideStringField;
        DSSet: TDataSource;
        DSMaster: TDataSource;
        DSDiv: TDataSource;
        TbDiv: TADOTable;
        TbDivid: TAutoIncField;
        TbDivDSDesigner: TWideStringField;
        TbDivDSDesigner2: TWideStringField;
        TbDivDSDesigner3: TWideStringField;
        TbComp: TADOTable;
        DSComp: TDataSource;
        TbCompid: TAutoIncField;
        TbCompid_: TIntegerField;
        TbCompDSDesigner: TWideStringField;
        TbCompDSDesigner2: TWideStringField;
```

Продолжение ПРИЛОЖЕНИЕ Б
Фрагменты исходного кода программы

```
TbCompDSDesigner3: TWideStringField;  
TbCompDSDesigner4: TWideStringField;  
TbCompIP: TWideStringField;  
TbCompDSDesigner5: TDateTimeField;  
TbCompDSDesigner6: TWideStringField;  
TbCompDSDesigner7: TWideStringField;  
TbSetid: TAutoIncField;  
TbSetid_: TIntegerField;  
TbSetid2: TIntegerField;  
TbSetid3: TIntegerField;  
TbSetid4: TIntegerField;  
TbSetDSDesigner: TWideStringField;  
TbSetDSDesigner2: TDateTimeField;  
TbSetDSDesigner3: TBCDField;  
TbSetDSDesigner4: TDateTimeField;  
TbSetField: TStringField;  
TbSetField2: TStringField;  
TbSetField3: TStringField;  
TbHistory: TADOTable;  
TbSProject: TADOTable;  
DSHistory: TDataSource;  
DSSProject: TDataSource;  
DSHistory1: TDataSource;  
TbHistory1: TADOTable;  
TbHistory1id: TAutoIncField;  
TbHistory1id_: TIntegerField;  
TbHistory1_: TDateTimeField;  
TbHistory1id_2: TIntegerField;  
TbHistory1DSDesigner2: TBooleanField;  
TbHistory1DSDesigner3: TBooleanField;  
TbHistory1DSDesigner4: TBooleanField;  
TbHistory1Field: TStringField;  
TbComp1: TADOTable;  
TbComp1id: TAutoIncField;  
TbComp1id_: TIntegerField;  
TbComp1DSDesigner: TWideStringField;  
TbComp1DSDesigner2: TWideStringField;  
TbComp1DSDesigner3: TWideStringField;  
TbComp1DSDesigner4: TWideStringField;  
TbComp1IP: TWideStringField;  
TbComp1DSDesigner5: TDateTimeField;  
TbComp1DSDesigner6: TWideStringField;  
TbComp1DSDesigner7: TWideStringField;  
DSComp1: TDataSource;  
TbProcess: TADOTable;  
DSProcess: TDataSource;  
TbWork: TADOTable;  
DSWork: TDataSource;  
TbSProjectid_: TAutoIncField;  
TbSProjectDSDesigner: TWideStringField;  
TbSProjectid: TIntegerField;  
TbSProjectDSDesigner2: TWideStringField;  
TbSProjectDSDesigner3: TDateTimeField;  
TbSProjectDSDesigner4: TDateTimeField;  
TbSProjectDSDesigner5: TBCDField;  
TbSProjectDSDesigner6: TWideStringField;  
TbSProjectField: TStringField;
```