

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Санкт-Петербургский политехнический университет Петра Великого»

Институт компьютерных наук и технологий
Кафедра «Системы и технологии управления»

Работа допущена к защите
Заведующий кафедрой СТУ
_____ В.П. Шкодырев
«__» _____ 2016 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

**Разработка автоматизированной системы учета и обслуживания технических и
программных средств малого предприятия**
направление 09.03.01 – Информатика и вычислительная техника
профиль 09.03.01_07 – Интеллектуальные системы обработки информации и
управления

Выполнил
студент гр. 43503/3

<

>

Р.С Колпаков

Научный руководитель
(к.т.н, доцент)

<

>

А.М Верхолат

Санкт-Петербург

2016

РЕФЕРАТ

Пояснительная записка к ВКР 60 с., 12 источников, 2 приложения, 16 рисунков, 14 таблиц.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА, РАЗРАБОТКА, БАЗА ДАННЫХ, ПРЕДМЕТНАЯ ОБЛАСТЬ, ИНВЕНТАРИЗАЦИЯ, ОБСЛУЖИВАНИЕ

Цель работы – разработка информационной системы учета и обслуживания технических и программных средств малого предприятия.

Разрабатываемая информационная система позволит упростить работу сотрудников отдела, на который возложены задачи учета и инвентаризации, автоматизируя такие процессы, как ведение документооборота, ведение учета и обслуживания доступ к информационным ресурсам и многое другое. Данная система обеспечит широкие возможности руководству отдела по автоматизированному управлению его деятельностью.

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ.....	3
Введение.....	4
1. Анализ предметной области. Постановка задачи.....	6
1.1 Описание предметной области.	6
1.2 Описание бизнес-логики.....	7
1.3 Группы пользователей.	17
1.4 Функциональные требования.....	18
1.5 Определение требований к проектируемой системе. Построение модели прецедентов.	19
2. Структурное построение информационной системы	25
2.1 Проектирование базы данных.	27
2.1.1 Инфологическая модель и описание сущностей.	27
2.1.2 Даталогическая модель.	33
2.2 Разработка пользовательского интерфейса.....	41
3. Выбор и обоснование инструментальных средств разработки и документирования.	43
3.1 Выбор операционной системы.....	43
3.2 Выбор средства для документирования.....	43
3.3 Выбор средств проектирования	43
3.4 Выбор СУБД. Проектирование серверной части.....	46
3.5 Выбор и обоснование языка программирования	49
4. Программная реализация автоматизированной информационной системы.....	50
Заключение.....	56
5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	57

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В данной работе применяются следующие термины с соответствующими определениями:

- АИС – автоматизированная информационная система;
- ЭВМ – электронно-вычислительная машина;
- БД – база данных;
- ИС – инфологическая схема;
- ДМ – даталогическая модель;
- ПО – предметная область;
- СУБД – система управления базами данных;
- Техническое средство – компьютерное или сопутствующее оборудование
- Программное средство – программное обеспечение, программные комплексы, пакеты прикладных программ.

Введение

Так или иначе, большому числу компаний приходится сталкиваться с задачей учета компьютерной техники, программного обеспечения и прочего оборудования. Существует несколько способов решения этой проблемы. Вариант первый – использовать для учета средства офисных приложений. Например, вести таблицу в MS Excel или другом табличном редакторе. Плюсы: тривиальное решение, требующее только навыков в использовании табличного редактора. Минусы: необходимость вводить все данные вручную, трудности при обновлении, отслеживании изменений в конфигурации оборудования.

Вариант второй – воспользоваться программой для автоматического сбора информации об аппаратной конфигурации компьютеров с последующим занесением в базу данных. Этот вариант имеет следующие преимущества: сбор конфигурации производится автоматически и требует лишь небольшого редактирования в дальнейшем. Хранение информации в формате базы данных значительно облегчает последующее построение отчетов. Автоматическое обновление информации о конфигурациях компьютеров позволяет повысить оперативность при работе с базой данных.

Целью данной выпускной квалификационной работы является разработка автоматизированной системы учета и контроля обслуживания технических и программных средств.

В соответствии с поставленной целью в работе определены следующие задачи:

1. проведение анализа предметной области;
2. выявление требований пользователей к информационной системе;
3. разработка структуры информационной системы и выделение основных функциональных блоков;

4. проектирование структуры базы данных (инфологическое, даталогическое и физическое моделирование);
5. разработка интерфейса пользователя;
6. программная реализация АИС

Актуальность разработки такой автоматизированной системы обусловлена тем, что в процессе функционирования отдела, решается перечень рутинных задач, при выполнении которых формируется множество документов долговременного хранения, оперативная работа с которыми возможна только в рамках автоматизированной информационной системы.

1. Анализ предметной области. Постановка задачи.

Анализ предметной области – это начальный шаг этапа системного анализа, с которого начинается разработка программного продукта.

Предметная область оказывает сильнейшее влияние на все аспекты проекта: взаимодействие с пользователем, требования к системе, используемая для хранения данных модель, реализация и многое другое.

Анализ предметной области проводится для выделения ее сущностей, определения первоначальных требований к функциональности и определения границ проекта [3].

1.1 Описание предметной области.

Отдел учета и обслуживания технических и программных средств занимается:

1. Учетом технических и программных средств, находящихся на рабочих местах пользователей.
2. Проведением инвентаризации технических и программных средств, в соответствии с регламентом.
3. Постановкой на учет нового оборудования.
4. Принятием заявок на проведение ремонтных работ и их реализацией.

Исходя из функций, выполняемых отделом, целесообразно создание автоматизированной информационной системы, позволяющей автоматизировать выполнение следующих задач, решаемых в отделе:

1. Автоматизация учета технических и программных средств находящихся на рабочих местах пользователей.
2. Автоматизация инвентаризации.
3. Автоматизации постановки на учет нового оборудования
4. Автоматизация приема заявок на обслуживание и контроль их реализации.
5. Автоматизация документооборота отдела.

1.2 Описание бизнес-логики.

Процесс бизнес-моделирования может быть реализован с применением различных методик, определяющим факторов для выбора той или иной методики является область деятельности моделируемого объекта, в моем случае это отдел учета и эксплуатации. Исходя из различных представлений о цели моделирования, принято делить методики на функциональные и объектные [5].

Методики объектного моделирования рассматривают моделируемый отдел как совокупность взаимодействующих объектов. Объект определяется как предмет или явление, имеющее четко определяемое поведение. Целью применения такой методики является разграничение объектов, составляющих отдел, и распределение между ними ответственностей за выполняемые действия.

Функциональные методики рассматривают моделируемый отдел как набор функций, преобразующих поток поступающей информации в выходной поток. Таким образом, в функционально-ориентированных моделях (IDEF – моделях, DFD – диаграммах потоков данных) главными структурными компонентами являются функции (операции, действия, работы), связываемые между собой потоками объектов.

Необходимо отметить несомненное достоинство функциональных моделей – они позволяют реализовать структурный подход к проектированию АИС по принципу "сверху – вниз", когда каждый функциональный блок может быть декомпозирован на множество отдельных блоков и так далее, выполняя, таким образом, модульное проектирование АИС. Для функциональных моделей характерны процедурная строгость декомпозиции ИС и наглядность представления.

Для описания бизнес-логики предметной области было использовано CASE – средство верхнего уровня BPwin. Это инструментальное средство системного анализа и проектирования информационных систем.

Модель, реализованная средствами BPwin, позволяет грамотно документировать необходимые аспекты деятельности – те действия, которые необходимо предпринять, способы их реализации, требующиеся для этого ресурсы.

BPwin совмещает в одном инструменте средства моделирования функций (IDEF0), потоков данных (DFD) и потоков работ (IDEF3) [6].

Функциональное моделирование (нотация IDEF0).

Наиболее подходящим языком моделирования бизнес-процессов в данном случае является IDEF0, первоначально называвшийся SADT – Structured Analysis and Design Technique. В IDEF0 система представляется совокупностью взаимодействующих между собой работ или функций. Выбранная функциональная ориентация является принципиальной, ведь функции системы анализируются вне зависимости от объектов, которыми они оперируют в системе. Это позволяет четче смоделировать логику взаимодействия процессов организации. В IDEF0 под моделью понимают описание системы (графическое и текстовое), способное дать ответ на некоторые заранее поставленные вопросы.

Функциональное моделирование позволяет провести системный анализ бизнеса, сосредоточив внимание на регулярных функциях (задачах), показателях, которые свидетельствуют об их правильном выполнении, ресурсах необходимых для этого, исходных материалах и результатах.

В IDEF0 система может быть представлена как совокупность взаимодействующих функций или работ. Система и её функции анализируются вне зависимости от объектов, которыми они оперируют. Это позволяет четче моделировать логику и взаимодействие процессов организации [6].

Сначала проводится описание системы в общем и её взаимодействие с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция – система разбивается на более компактные подсистемы и каждая подсистема описывается отдельно (диаграммы

декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее пока не будет достигнута нужная степень подробности. Такая технология создания модели позволяет построить модель, адекватную предметной области на всех уровнях абстрагирования.

Функциональные модели представлены на рисунках 1-3.

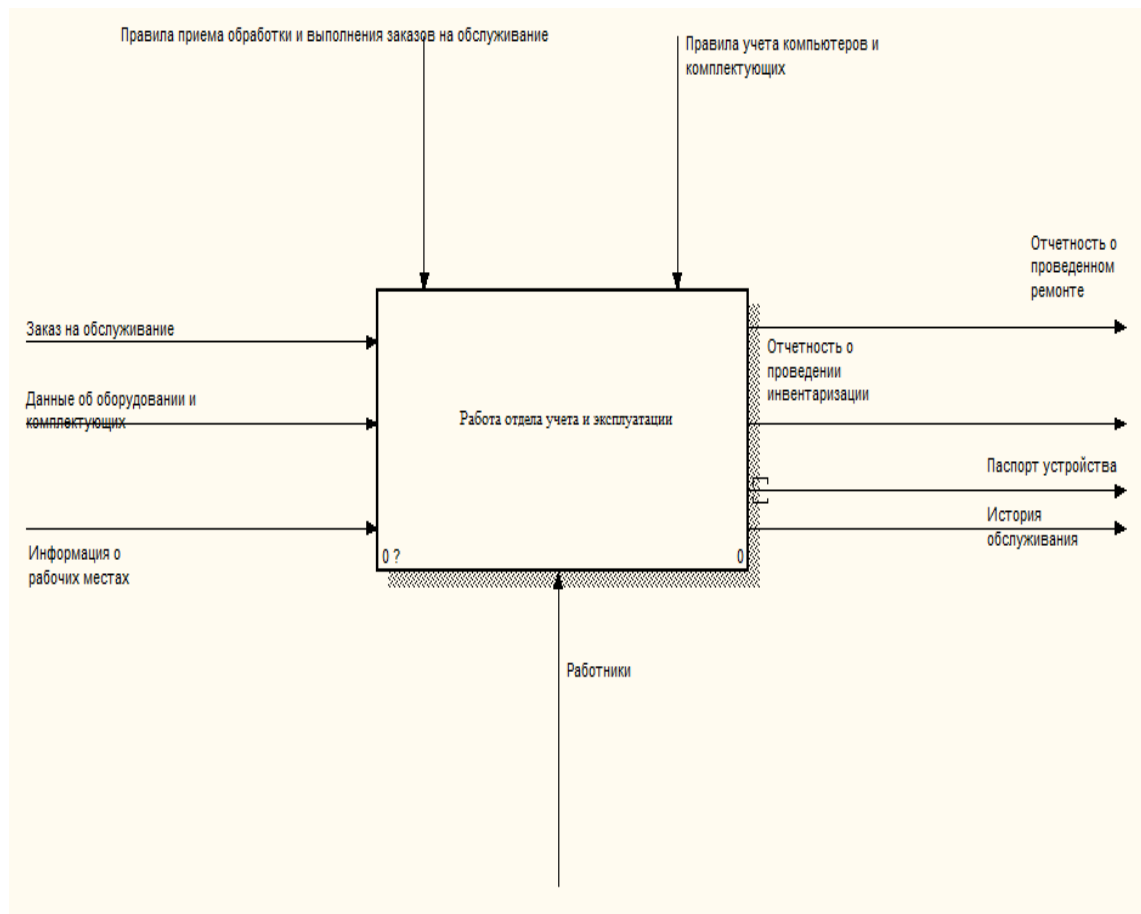


Рисунок 1 - I уровень IDEF0

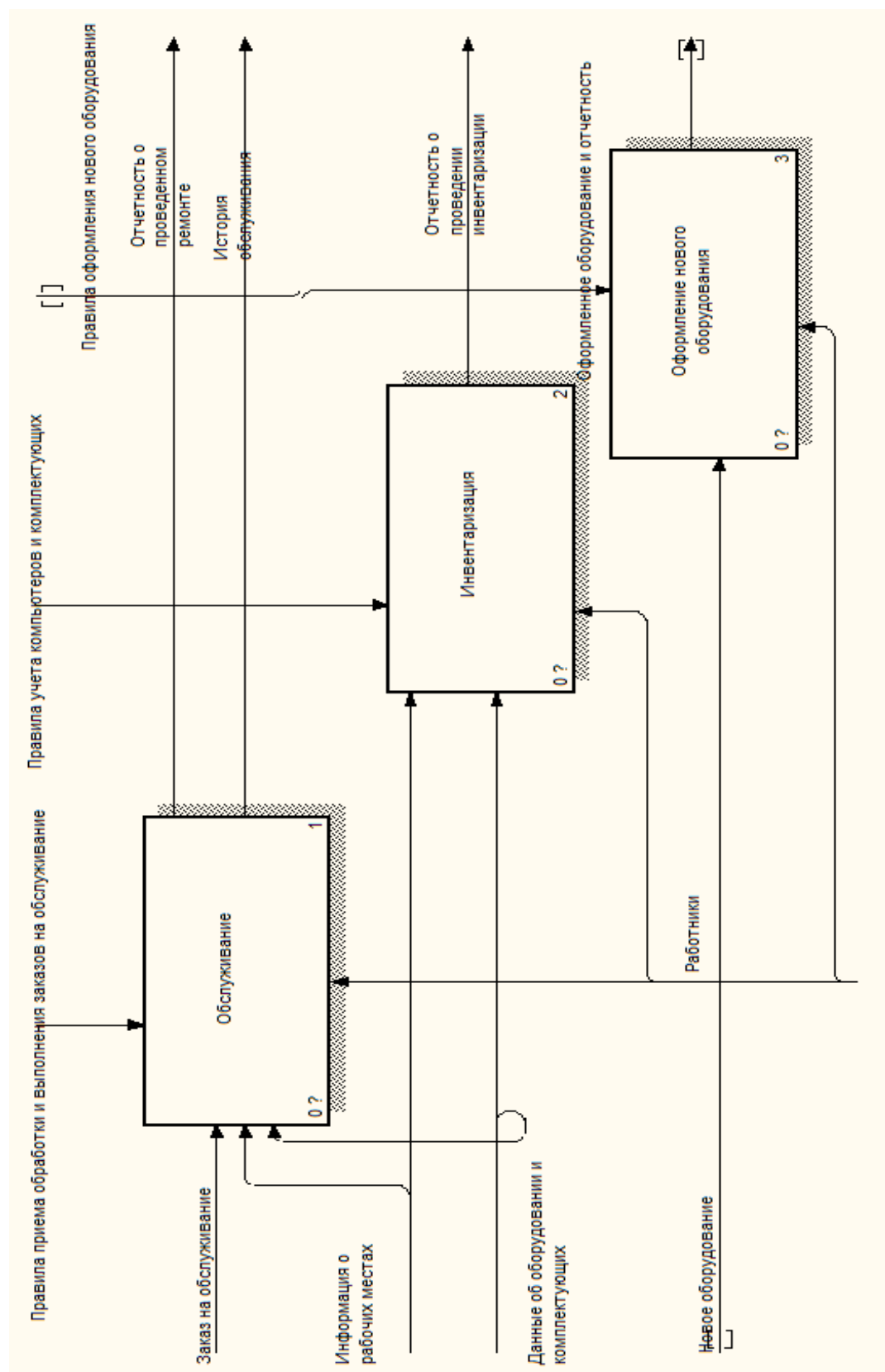


Рисунок 2- II уровень IDEF0

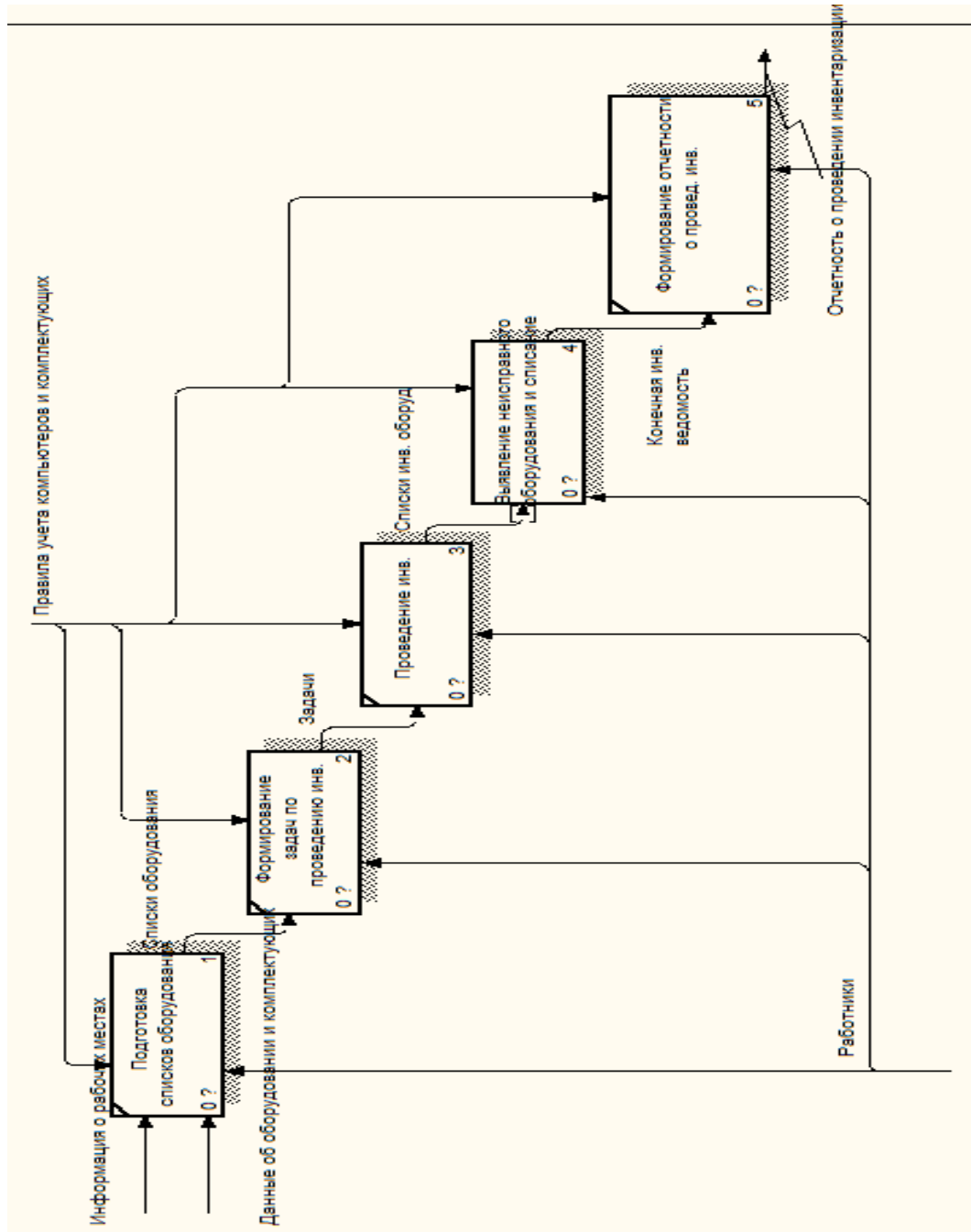


Рисунок 4- III уровень IDEF0 (Инвентаризация)

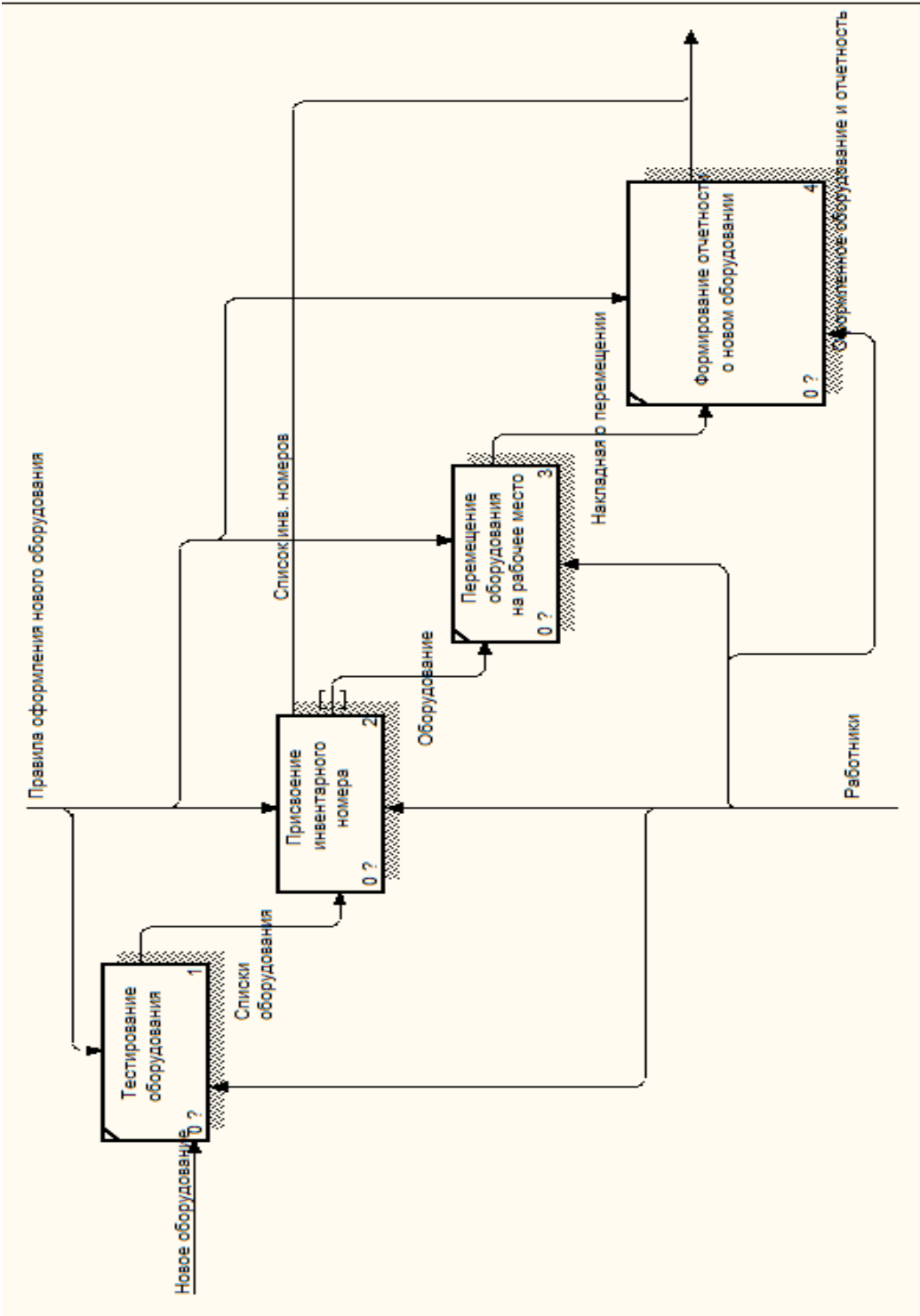


Рисунок 5- III уровень IDEF0 (Оформление нового оборудования)

Моделирование потоков данных (нотация DFD).

Диаграммы потоков данных (DFD) используются для описания документооборота и потоков обработки информации.

Диаграммы потоков данных (DFD, Data flow diagramming) используются для описания документооборота и обработки информации в исследуемой сфере. Подобно IDEF0, DFD представляет модельную систему как сеть связанных между собой работ. Их удобно использовать как дополнение к модели IDEF0 для более наглядного отображения существующих операций документооборота в корпоративных системах обработки информации.

Моделирование потоков данных, часто используемое при разработке программного обеспечения, сосредотачивается вокруг потоков данных, передающихся между различными задачами, включая их хранение, с целью достичь максимальной доступности и минимизировать время ответа. Подобное моделирование позволяет рассматривать конкретный процесс, анализировать операции, из которых он состоит, а также точки принятия решений, влияющих на его ход.

DFD описывает:

- функции обработки информации (работы);
- документы (стрелки), объекты, которые участвуют в обработке информации;
- внешние ссылки, которые обеспечивают интерфейс с внешними объектами;
- таблицы для хранения документов (хранилище данных).

Стрелки DFD показывают, как объекты (включая данные) двигаются от одной работы к другой.

Диаграммы потоков данных представлены на рисунках 6-8.

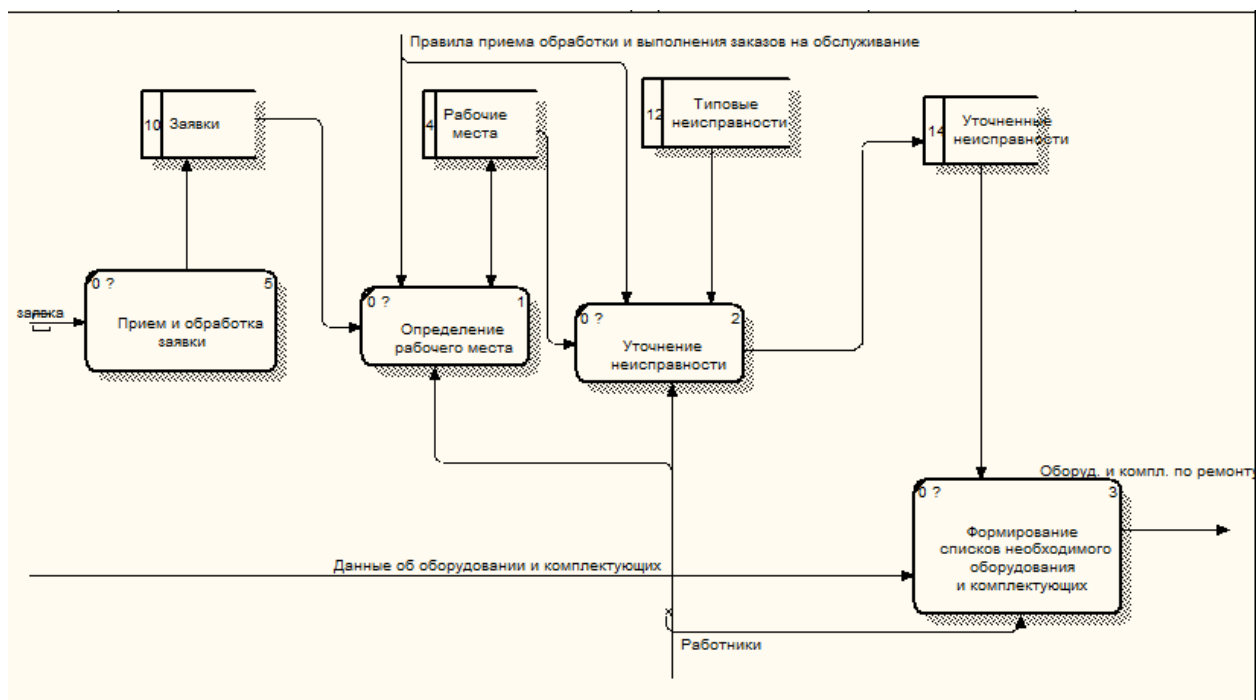


Рисунок 6 - III уровень DFD (Уточнение ремонтных работ)

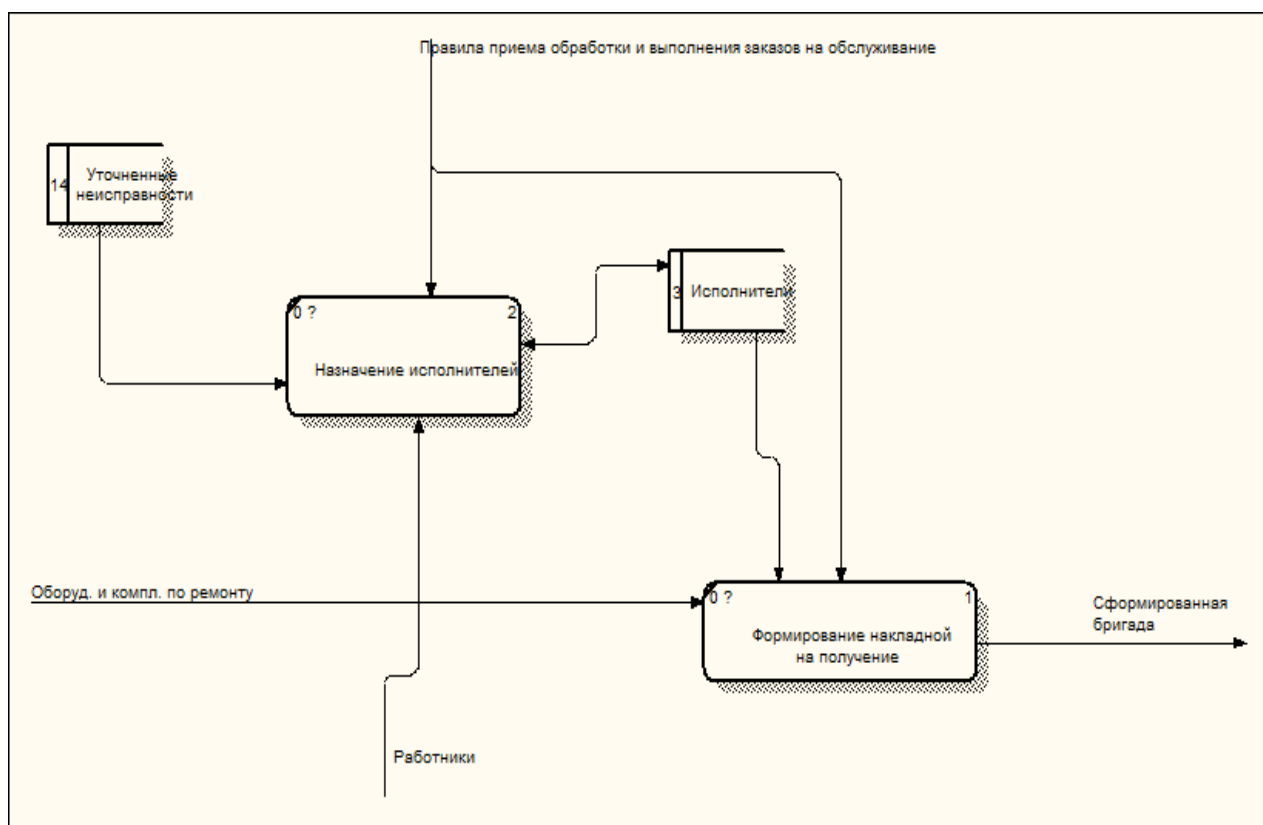


Рисунок 7 - III уровень DFD (Формирование ремонтной бригады)

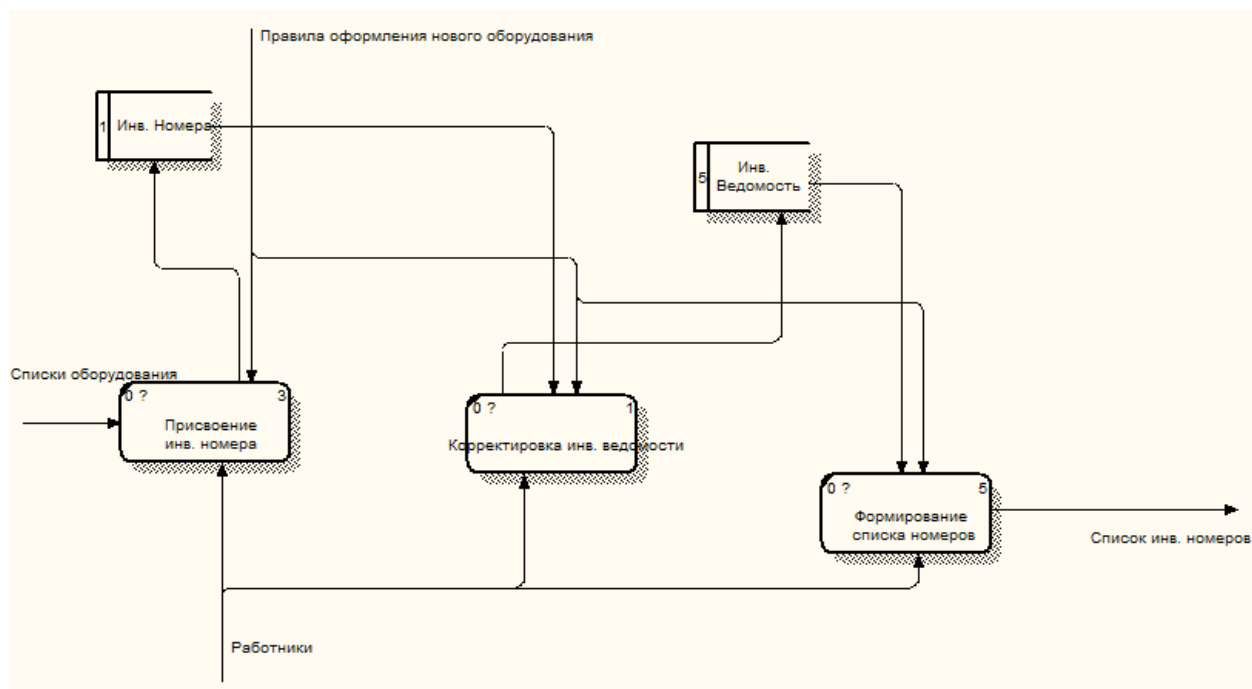


Рисунок 8 - III уровень DFD (Присвоение инв. номера)

1.3 Группы пользователей.

В процессе анализа предметной области были сформированы профили пользователей разрабатываемой информационной системы. Под профилем понимается описание главных категорий пользователей. Для разрабатываемой информационной системы выделен один класс потенциальных пользователей – сотрудники отдела учета и эксплуатации.

В качестве способа получения информации был выбран опрос целевой аудитории. Описание профиля включает в себя такие разделы, как:

- социальные характеристики;
- навыки и умения работы с компьютером и данной системой;
- мотивационно-целевая среда;
- рабочая среда;
- особенности взаимодействия с компьютером (специфические требования пользователей, необходимые информационные технологии).

В таблице 1 представлен профиль пользователей системы.

Таблица 1 - Профиль пользователей системы

Пользователь	Сотрудник отдела
Социальные характеристики	Мужчины, женщины, взрослые (старше 18 лет), владеют русским языком, уровень владения компьютером зависит от поставленных должностных задач.
Навыки и умения	Самостоятельного ознакомления достаточно для выполнения всех необходимых задач.
Мотивационно-целевая среда	Удобство, мобильность, мотивация к обучению высокая.
Рабочая среда	Необходимое программное обеспечение и оборудование.
Требования к разрабатываемому	Возможность наглядного представления и редактирования данных, доступ к информационным

приложению	ресурсам отдела, ведение документооборота.
------------	--

1.4 Функциональные требования.

Необходимо разработать автоматизированную информационную систему (АИС) учета и обслуживания оборудования и комплектующих, способную выполнять следующие основные функции:

- учет оборудования (компьютерной техники) по автоматизированным рабочим местам, рабочих мест по отделам, производителей и поставщиков оборудования
- оформление и учет прихода оборудования;
- перемещение оборудования с одного рабочего места пользователя на другое;
- учет заявок на ремонт или обслуживание оборудования и оргтехники подразделений организации;
- ведение истории обслуживания автоматизированного рабочего места;
- реализация информационно-справочных функций системы (быстрый поиск по оборудованию, и документам; возможность сортировки и фильтрации информации для удобства ее восприятия).
- возможность задавать для каждого комплектующего произвольный набор характеристик и параметров;
- формирование следующей отчетности:
 - Паспорта выбранного устройства;
 - списка инвентарных номеров;
 - списка компьютерной техники по рабочим местам пользователей;
 - накладной на внутреннее перемещение оборудования и комплектующих;
 - списка рабочих мест;
 - списка оборудования на выбранном рабочем месте;
 - истории обслуживания выбранного рабочего места;
 - истории перемещения выбранного комплектующего.

Автоматизированная информационная система должна быть реализована в рамках клиент-серверной архитектуры.

Система должна обладать удобным интерфейсом, предоставляющим наиболее гибкий способ просмотра, добавления, редактирования и удаления данных.

1.5 Определение требований к проектируемой системе. Построение модели прецедентов.

Одним из способов представления требований пользователей являются диаграммы вариантов использования (диаграммы прецедентов, use case diagrams), являющиеся частью языка UML.

UML — язык графического описания для создания объектных моделей в области разработки программного обеспечения. UML считается языком широкого профиля, он представляет собой открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, именуемой UML моделью. UML был разработан для определения, визуализации, проектирования и документирования программных систем. Основная задача диаграммы вариантов использования — представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы. Цель этого общения — выявить и формализовать требования к создаваемой системе.

Сложность разработки программного комплекса заключается во взаимодействии с пользователем, обработке его запросов, контроле корректности и безопасности. Применение диаграммы вариантов использования позволяет выявить ту функциональность, с помощью которой пользователи будут выполнять конкретные задачи. Так удастся избежать появления функций, которые в ходе сбора информации могут казаться весьма полезными, однако которые не будут использоваться из-за того, что они не связаны напрямую с решением рабочих задач.

Диаграмма вариантов использования заключается в следующем: создаваемая система представляется в виде некоего множества сущностей или актеров, взаимодействующих с проектируемой системой с помощью вариантов использования. При этом действующим лицом или актером называется любая сущность, осуществляющая взаимодействие с системой извне. Это может быть техническое устройство, человек, любая другая программа или система, которая может оказывать воздействие на моделируемую систему. В свою очередь, вариант использования (use case) служит для описания сервисов, предоставляемых системой актеру. Иначе говоря, каждый из вариантов использования определяет некоторый набор действий, которые совершаются системой при диалоге с актером. При этом ничего не говорится о том, каким способом реализуется взаимодействие актеров с системой [7].

На рисунке 9 представлена диаграмма вариантов использования для зарегистрированного пользователя.

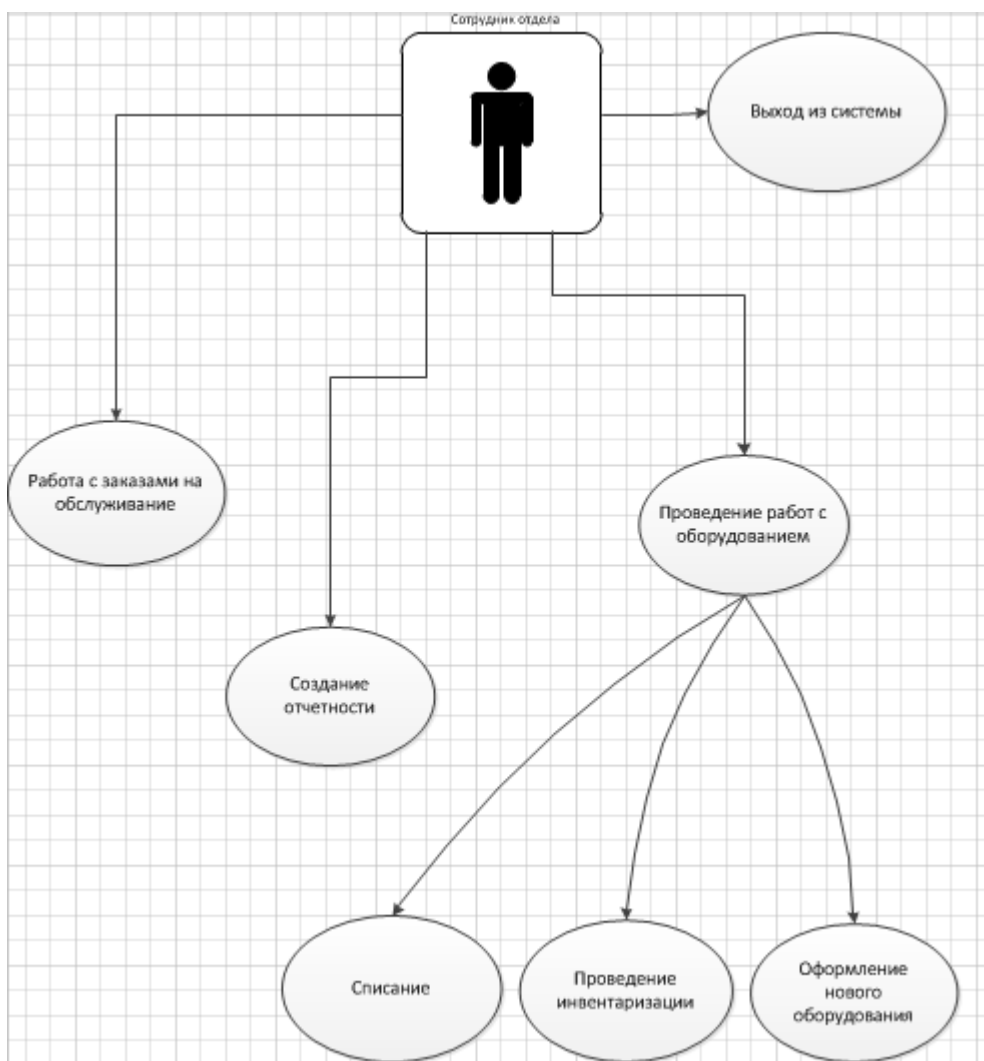


Рисунок 9 - Диаграммы вариантов использования для зарегистрированного пользователя

Описание вариантов использования представлено в таблицах 2-10.

Таблица 2 - Вариант использования - Оформление нового оборудования

Название	Оформление нового оборудования
Описание	Учет закупленного оборудования.
Частота использования	Является часто используемым.
Входные условия	Данные поставщика.
Выходные условия	Закупленное оборудование
Нормальное направление	<p>1.0 Поставки оборудования</p> <p>1. Пользователь запрашивает требующуюся информацию по имеющимся заказам или товарам поставщика для отображения, изменения, удаления. Вводит информацию по новым заказам и товарам.</p> <p>2. Выполняет запрос к БД</p> <p>3. Отображается, изменяется или удаляется</p>

	запрашиваемая информация. Добавляется новая информация.
Альтернативные направления	Нет
Исключения	1.0.И.1 1. Выводится сообщение об ошибке, если введены некорректные данные.

Таблица 3 - Вариант использования - Списание

Название	Списание
Описание	Отображение информации по списанному оборудованию, списание оборудования.
Частота использования	Является используемым
Входные условия	Данные об оборудовании
Выходные условия	Решение о списании
Нормальное направление	1. Пользователь запрашивает требующуюся информацию для отображения, изменения, удаления. Вводит новую информацию. 2. Выполняет запрос к БД 3. Отображается, изменяется или удаляется запрашиваемая информация. Добавляется новая информация.
Альтернативные направления	Нет
Исключения	Нет

Таблица 4 - Вариант использования - Проведение инвентаризации

Название	Проведение инвентаризации
Описание	Проведение инвентаризации
Частота использования	Является используемым.
Входные условия	Списки оборудования
Выходные условия	Списки инв. Оборудования
Нормальное направление	1. Пользователь запрашивает требующуюся информацию для отображения, изменения, удаления. Вводит новую информацию. 2. Выполняет запрос к БД 3. Отображается, изменяется или удаляется запрашиваемая информация.

	Добавляется новая информация.
Альтернативные направления	1. Нет
Исключения	1.0.И.1Отсутствие введенных данных

Таблица 5 - Вариант использования - Выход

Название	Выход
Описание	Завершение работы с данными информационной системы.
Частота использования	Является часто используемым.
Входные условия	Нет
Выходные условия	Нет
Нормальное направление	1.0 Выход 1. Пользователь выбирает пункт меню «Выход»; 2. Завершение сеанса пользователя;
Альтернативные направления	Нет.
Исключения	Нет.

Таблица 6 - Вариант использования - Создание отчетности

Название	Создание отчетности
Описание	Создание выходной отчетности о проведение инвентаризации, списании, добавлении нового оборудования, перемещении существующего
Частота использования	Является часто используемым.
Входные условия	Данные об оборудовании, перемещении, закупке, инвентаризации.
Выходные условия	Созданная отчетность.
Нормальное направление	1. Пользователь запрашивает требующуюся информацию по отчетности для отображения.. 2. Выполняет запрос к БД 3. Отображается, запрашиваемая информация.
Альтернативные направления	Нет.
Исключения	Нет.

Таблица 7 - Вариант использования - Проведение работ с оборудованием

Название	Проведение работ с оборудованием
Описание	Проведение работ с оборудованием
Частота использования	Является часто используемым.
Входные условия	Данные об оборудовании
Выходные условия	Данные по оборудованию., расположение оборудования, количество.
Нормальное направление	<ol style="list-style-type: none"> 1. Пользователь запрашивает информацию по оборудованию. 2. Выполняет запрос к БД 3. Отображается список оборудования их местоположение, характеристики. 4. Добавляется или изменяется информация по имеющемуся оборудованию.
Альтернативные направления	Нет.
Исключения	Нет.

Таблица 8 - Вариант использования - Работа с заказами на обслуживание

Название	Работа с заказами на обслуживание
Описание	Отображение информации по заказу на обслуживание. Добавление заказа или редактирование существующего заказа.
Частота использования	Является очень часто используемым
Входные условия	Выбранный заказ
Выходные условия	Нет
Нормальное направление	<ol style="list-style-type: none"> 1. Пользователь запрашивает требующуюся информацию по заказу на обслуживание для редактирования или, если заказ еще не оформлен, то вводит данные для добавления. 2. Выполняет запрос к БД 3. Отображается, изменяется или удаляется запрашиваемая информация. Добавляется новая информация.
Альтернативные направления	Нет
Исключения	Нет

2. Структурное построение информационной системы

2.1 Структура системы

В результате анализа и конкретизации предметной области, анализа требований пользователей была сформирована обобщенная структура построения комплекса, которая приведена на рисунке 10.

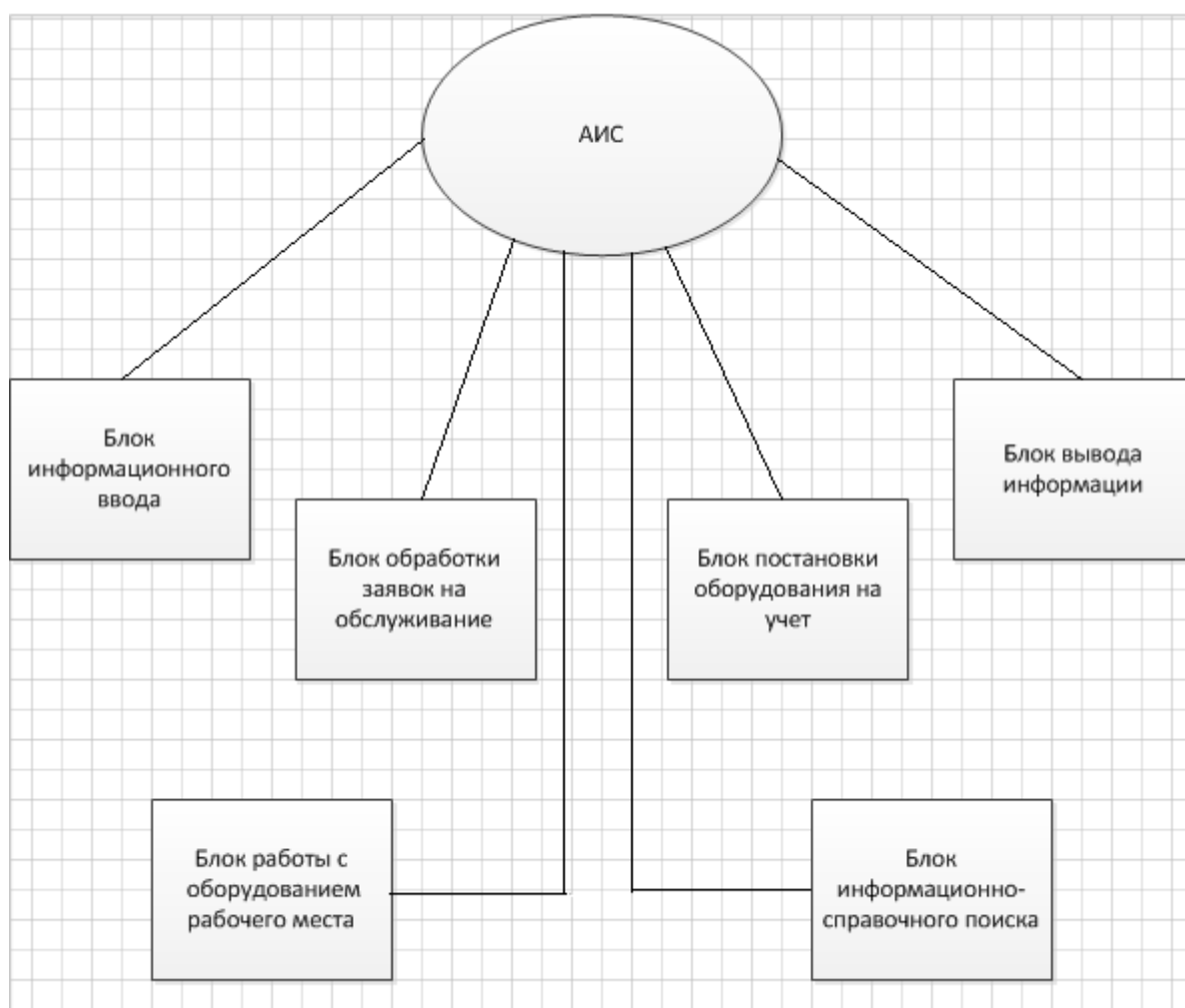


Рисунок 10 – Структура системы

Блок работы со справочной информацией включает в себя следующие справочники:

- типы устройств;
- параметры устройств;

- сотрудники IT-отдела;
- поставщики;
- производителя;
- состояния заказа.

Назначением данного модуля является поиск и предоставление пользователю справочной информации.

Блок информационного ввода используется для ввода первичных данных и просмотра ранее занесенных.

Блок обработки заявок на обслуживание выполняет задачи ввода заявок по обслуживанию и ремонту компьютерной техники.

Блок работы с оборудованием рабочего места выполняет задачи по формированию номенклатуры компьютерной техники и работы с ней, с помощью разработанных форм.

Блок постановки оборудования на учет выполняет задачи по оформлению нового оборудования.

Блок вывода информации используется для формирования печатных форм. Отчеты формируются, с использованием запросов, которые обрабатывают исходную информацию в соответствии с заданными параметрами пользователя.

2.1 Проектирование базы данных.

Для проектирования структуры базы данных было выбрано средство Erwin.

2.1.1 Инфологическая модель и описание сущностей.

Цель инфологического проектирования – обеспечить наиболее естественный для человека способ сбора и представления информации, хранимой в проектируемой базе данных. Из-за этого модель данных нужно строить аналогично естественному языку (естественный язык, однако, нельзя использовать в чистом виде из-за проблемы с компьютерной обработкой и неоднозначности понятий естественного языка). В основе модели лежат сущности их отношения и атрибуты.

Сущность – это любой отличимый объект (объект, который мы способны отличить), информация о котором должна храниться в базе данных. Логическая структура базы данных – это описание типа, длины и состава информационных структур базы данных, а так же их связи.

Связи и сущности модели данных обычно представляются в виде отношения (реляционной таблицы). Отношение, которое соответствует сущности, содержит столбцы (атрибуты), которые являются атрибутами сущности и описывающие сущность (объект). Атрибут или множество атрибутов, однозначно определяющие объект именуется ключом..

Можно представить отношение как таблицу, в которой каждая строка это кортеж, и каждый столбец сопоставляется одному компоненту. Столбцы называются атрибутами и им присваиваются имена. В свою очередь, список имён атрибутов носит название схемы отношения. Совокупность таких схем отношений, которые используются для представления информации, носят название схемы базы данных, а их текущие значения – базой данных.

Построение инфологической модели можно разбить на следующие шаги:

- выявление сущностей;
- выявление зависимости между сущностями;
- выбор альтернативных и первичных ключей;
- выявление атрибутов сущности(-ей);
- приведение модели к необходимому уровню нормальной формы.

Задача инфологического этапа моделирования состоит в получении семантических (смысловых) моделей, которые отражают информационное содержание конкретной предметной области [9]. В качестве источников для проектирования используется ПО и информационные потребности пользователя.

Для описания инфологической модели используются ER–диаграммы: «сущность – связь».

Описание предметной области подразумевает рассмотрение ее в виде совокупности сущностей связанных отношениями, которые имеют набор атрибутов или свойств, характеризующих конкретно эту сущность. Группы всех похожих сущностей, объединенные вместе, образуют объект, внутри которого фиксируются только необходимые для создаваемой информационной системы свойства. Связь представляет собой ни что иное, как отношение между двумя объектами.

Объект обязан иметь один или несколько ключевых атрибутов, то есть атрибутов, уникально определяющих экземпляр объекта среди прочих подобных экземпляров объектов этого типа.

Совокупность связанных объектов определяет структуру рассматриваемой предметной области, которая используется для построения инфологической модели.

Каждому объекту необходимо сопоставить экземпляр объекта, который в свою очередь образован совокупностью конкретных значений атрибутов, и однозначно определяющийся с помощью ключа.

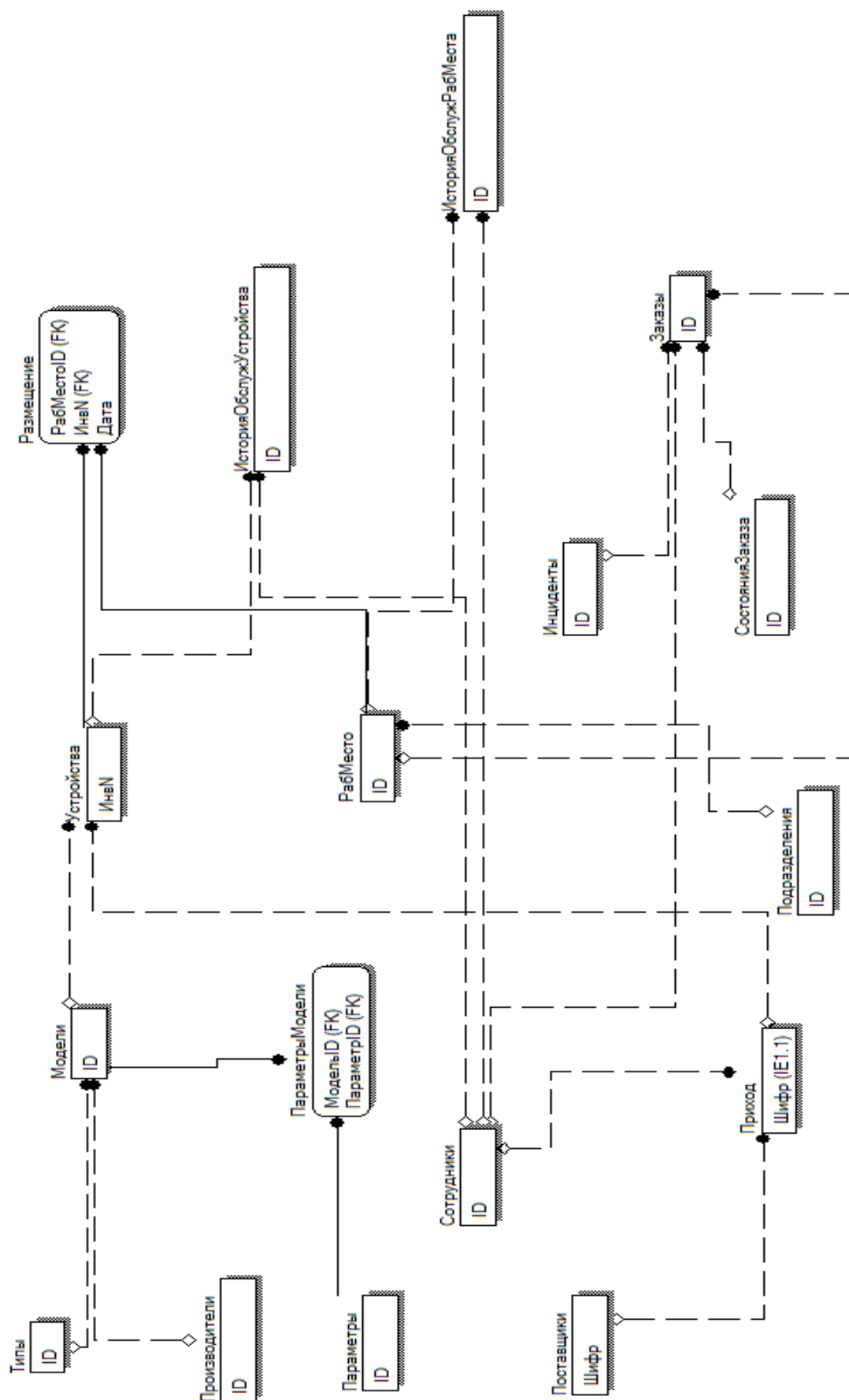


Рисунок 12 - Инфологическая схема (Уровень ключей)

Рисунок 13 - Инфологическая схема (Уровень атрибутов)

2.1.2 Даталогическая модель.

Глобальная даталогическая модель (ДМ) является отражением всего содержания БД, структурированного на логическом уровне, ориентированного на выбранную СУБД. Любая СУБД может оперировать только с допустимыми для нее логическими структурами. Все ДМ различны - отличаются наименования используемых информационных единиц, правила композиции структур (структуры более высокого уровня состоят из структур низкого уровня) и возможностью просмотра модели. Любая СУБД накладывает некоторые количественные ограничения на логическую структуру БД, а это в свою очередь, естественно оказывает некое влияние на проект ДМ. Поэтому прежде чем начать разработку даталогической модели надо, как можно более тщательно изучить выбранную СУБД, ознакомиться с ее ограничениями, изучить существующие методики проектирования в конкретной СУБД и определить основные факторы, оказывающие влияние на выбор логических структур данных [9].

Даталогическая схема представляет собой набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Разработанная даталогическая модель ориентирована на СУБД MS SQL Server 2014.

Спецификация ключей.

В таблице 12 представлена спецификация первичных ключей. В первых двух столбцах указывается соответственно имя и наименование ключа. Далее в столбцах указывается, является ли ключ первичным и мигрировал ли он из другой таблицы.

Entity Reports : Ключи (C:\Users\RK\Desktop\Диплом\Модель.erwin, 12:33:53, 20 rows)			
Entity Name	Entity Primary Key Attribute Name	Entity Primary Key Attribute Is PK	Entity Primary Key Attribute Is FK
Заказы	ID	Yes	No
Инциденты			
ИсторияОбслужРабМеста			
ИсторияОбслужУстройства			
Модели			
Параметры			
ПараметрыМодели	МодельID	Yes	Yes
	ПараметрID	Yes	Yes
Подразделения	ID	Yes	No
Поставщики	Шифр	Yes	No
Приход			
Производители	ID	Yes	No
РабМесто			
Размещение	РабМестоID	Yes	Yes
	ИнвN	Yes	Yes
	Дата	Yes	No
СостоянияЗаказа	ID	Yes	No
Сотрудники			
Типы			
Устройства	ИнвN	Yes	No

Таблица 12 - Спецификация ключей

Спецификация атрибутов.

В таблице 15 представлена спецификация атрибутов. В первом столбце – название таблицы, во втором – названия полей, в третьем – тип данных.

Таблица 13 - Спецификация атрибутов

Таблица	Поле	Тип
Заказы	ID	AutoNumber
	ДатаВремяПриема	Date/Time
	СостояниеID	AutoNumber
	ИнцидентID	
	РабМестоID	
	ДатаВремяВыпНач	Date/Time
	ДатаВремяВыпКон	
	ИсполнительID	AutoNumber
	Текст	Text(255)
	Подал	Text(50)
Инциденты	ID	AutoNumber

	Инцидент	Text(15)
ИсторияОбслужРабМеста	ID	AutoNumber
	РабМестоID	
	Дата	Date/Time
	Тип	Text(50)
	СотрудникID	AutoNumber
	Описание	Text(255)
ИсторияОбслужУстройства	ID	AutoNumber
	ИнвN	
	Дата	Date/Time
	Тип	Text(50)
	СотрудникID	AutoNumber
	Описание	Text(255)
Модели	ID	AutoNumber
	ТипID	
	Наименование	Text(255)
	ПроизводительID	AutoNumber
	Профилактика	Text(15)
	Амортизация	
Параметры	ID	AutoNumber
	Наименование	Text(100)
ПараметрыМодели	МодельID	AutoNumber
	ПараметрID	
	Значение	Text(100)
Подразделения	ID	AutoNumber
	Наименование	Text(50)
Поставщики	Шифр	AutoNumber
	Наименование	Text(100)
	Адрес	
	Телефоны	Text(30)
	Реквизиты	Text(100)
	ИНН	Text(15)
	КПП	
Приход	Шифр	AutoNumber
	ПоставщикID	
	Счет	Text(5)
	СчетДата	Date/Time
	СчетФактура	Text(5)
	СчетФактураДата	Date/Time

	Накладная	Text(5)
	НакладнаяДата	Date/Time
	СотрудникID	AutoNumber
Производители	ID	AutoNumber
	Производитель	Text(50)
	Сайт	Text(100)
РабМесто	ID	AutoNumber
	ПодразделениеID	AutoNumber
	Наименование	Text(255)
	СетевоеИмя	Text(50)
	IPАдрес	Text(15)
	Домен	
	ПО	Text(255)
	Описание	
	Телефон	Text(10)
Размещение	РабМестоID	AutoNumber
	ИнвN	
	Дата	Date/Time
СостоянияЗаказа	ID	AutoNumber
	Состояние	Text(50)
Сотрудники	ID	AutoNumber
	ФИО	Text(30)
	Должность	Text(50)
Типы	ID	AutoNumber
	Наименование	Text(50)
Устройства	ИнвN	AutoNumber
	МодельID	AutoNumber
	СерийныйN	Text(25)
	Дата	Date/Time
	ПриходID	AutoNumber
	Цена	Currency
	ДатаСписания	Date/Time

Даталогическая схема.

На рисунке 14 представлена даталогическая схема.

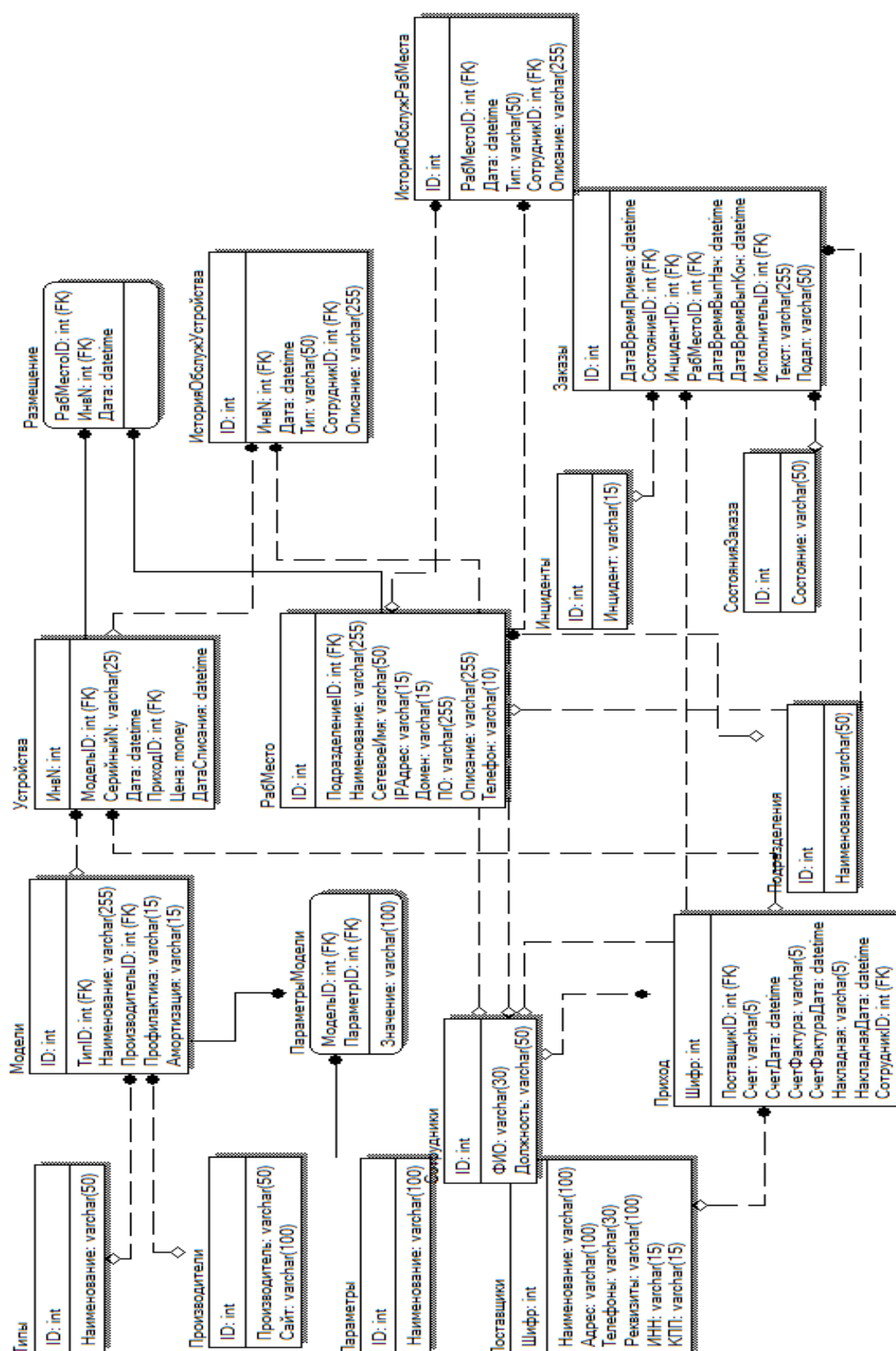


Рисунок 14- Даталогическая схема

Обеспечение целостности данных

В реляционной модели данных существует два базовых требования целостности, поддерживаемые любой реляционной СУБД. Первое требование именуется требованием целостности сущностей. Сущности или объекту реального мира в реляционных базах данных сопоставляются кортежи отношений. Конкретное требование состоит в следующем - любой кортеж любого отношения должен быть отличим от любого другого кортежа этого отношения, то есть другими словами, любое отношение обязано обладать первичным ключом.

Второе требование именуется требованием ссылочной целостности и является более сложным. Требование ссылочной целостности, или требование целостности по ссылкам заключается в следующем: для всякого значения внешнего ключа, возникающего в ссылающемся отношении, в отношении, к которому ведет ссылка, обязан существовать кортеж с таким же значением первичного ключа, либо значение внешнего ключа должно быть неопределенным (то есть ни на что не указывать).

Оба ограничения и ограничения и по ссылкам и ограничение целостности сущности должны поддерживаться СУБД. Для соблюдения целостности сущности достаточно соблюдать отсутствие в любом отношении кортежей с одним и тем же значением первичного ключа. С целостностью по ссылкам дела обстоят несколько сложнее.

При обновлении ссылающегося отношения (вставке новых кортежей или модификации значения внешнего ключа в существующих кортежах) достаточно следить за тем, чтобы не появились некорректные значения внешнего ключа.

При удалении кортежа из отношения, на которое ведет ссылка, существуют три подхода, поддерживающих целостность по ссылкам. Первый подход заключается в том, что запрещено производить удаление кортежа, на который существуют ссылающиеся объекты (то есть вначале нужно либо

удалить ссылающиеся кортежи, либо подходящим образом изменить значения их внешнего ключа). При втором подходе при удалении кортежа, на который ссылается другой кортеж, во всех ссылающихся кортежах значение внешнего ключа становится неопределенным. Наконец, третий подход (каскадное удаление) заключается в следующем - при удалении кортежа из отношения, на которое указывает ссылка, из ссылающегося отношения удаляются все ссылающиеся кортежи. В современных реляционных СУБД обычно можно выбрать удобный способ поддержания целостности по ссылкам для каждой отдельной ситуации определения внешнего ключа. Конечно, для принятия такого решения нужно анализировать требования исследуемой области.

В таблице 14 представлено обеспечение целостности данных.

Таблица 14 - Обеспечение целостности данных

Родительская сущность	Дочерняя таблица	Тип отношения	Изменение	Удаление
Инциденты	Заказы	Zero-or-One-to-Zero-One-or-More	Cascade	No Action
Модели	ПараметрыМодели	One-to-One-or-More (P)	Cascade	Cascade
	Устройства	Zero-or-One-to-Zero-One-or-More	Cascade	No Action
Параметры	ПараметрыМодели	One-to-One-or-More (P)	Cascade	No Action
Подразделения	РабМесто	Zero-or-One-to-Zero-One-or-More	No Action	Cascade
Поставщики	Приход	Zero-or-One-to-Zero-One-or-More	Cascade	No Action
Приход	Устройства	Zero-or-One-to-Zero-One-or-More	No Action	No Action
Производители	Модели	Zero-or-One-to-Zero-One-or-More	Cascade	No Action
РабМесто	Заказы	Zero-or-One-to-Zero-One-or-More	Cascade	No Action
	ИсторияОбслужРабМеста	Zero-or-One-to-Zero-One-or-More	Cascade	No Action
	Размещение	One-to-One-or-More (P)	Cascade	Cascade
СостоянияЗаказа	Заказы	Zero-or-One-to-Zero-One-or-More	Cascade	Cascade
Сотрудники	Заказы	Zero-or-One-to-Zero-One-or-More	Cascade	No Action
	ИсторияОбслужРабМеста	Zero-or-One-to-Zero-One-or-More	Cascade	No Action
	ИсторияОбслужУстройства	Zero-or-One-to-Zero-	Cascade	No Action

		One-or-More		
	Приход	Zero-or-One-to-Zero-One-or-More	Cascade	No Action
Типы	Модели	Zero-or-One-to-Zero-One-or-More	Cascade	No Action
Устройства	ИсторияОбслужУстройства	Zero-or-One-to-Zero-One-or-More	Cascade	No Action
	Размещение	One-to-One-or-More (P)	Cascade	Cascade

Noaction – Запрещается удаление записи, если существуют зависимые записи с соответствующим значением внешнего ключа.

Cascade – Удаление записи приводит к удалению записей с соответствующим значением внешнего ключа.

2.2 Разработка пользовательского интерфейса.

Графический интерфейс пользователя является средой для работы пользователя с данными, поэтому его разработка является одним из самых важных моментов написания программного обеспечения [10].

Графический пользовательский интерфейс в вычислительной технике – система, для обеспечения взаимодействия пользователя и компьютера, основывающаяся на представлении всего перечня системных объектов и функций, доступных пользователю, в лаконичном виде графических компонентов экрана (кнопок, окон, списков, полей для ввода и прочих).

В отличие от интерфейса командной строки, широко использующегося в прошлом, пользователь имеет произвольный доступ ко всем видимым экранным объектам при помощи клавиатуры или устройства координатного ввода (например, мыши).

Очень важно создать гибкий дружелюбный и лаконичный интерфейс, приемлемый и понятный пользователю.

Ниже, представлен пример структуры пользовательского интерфейса:

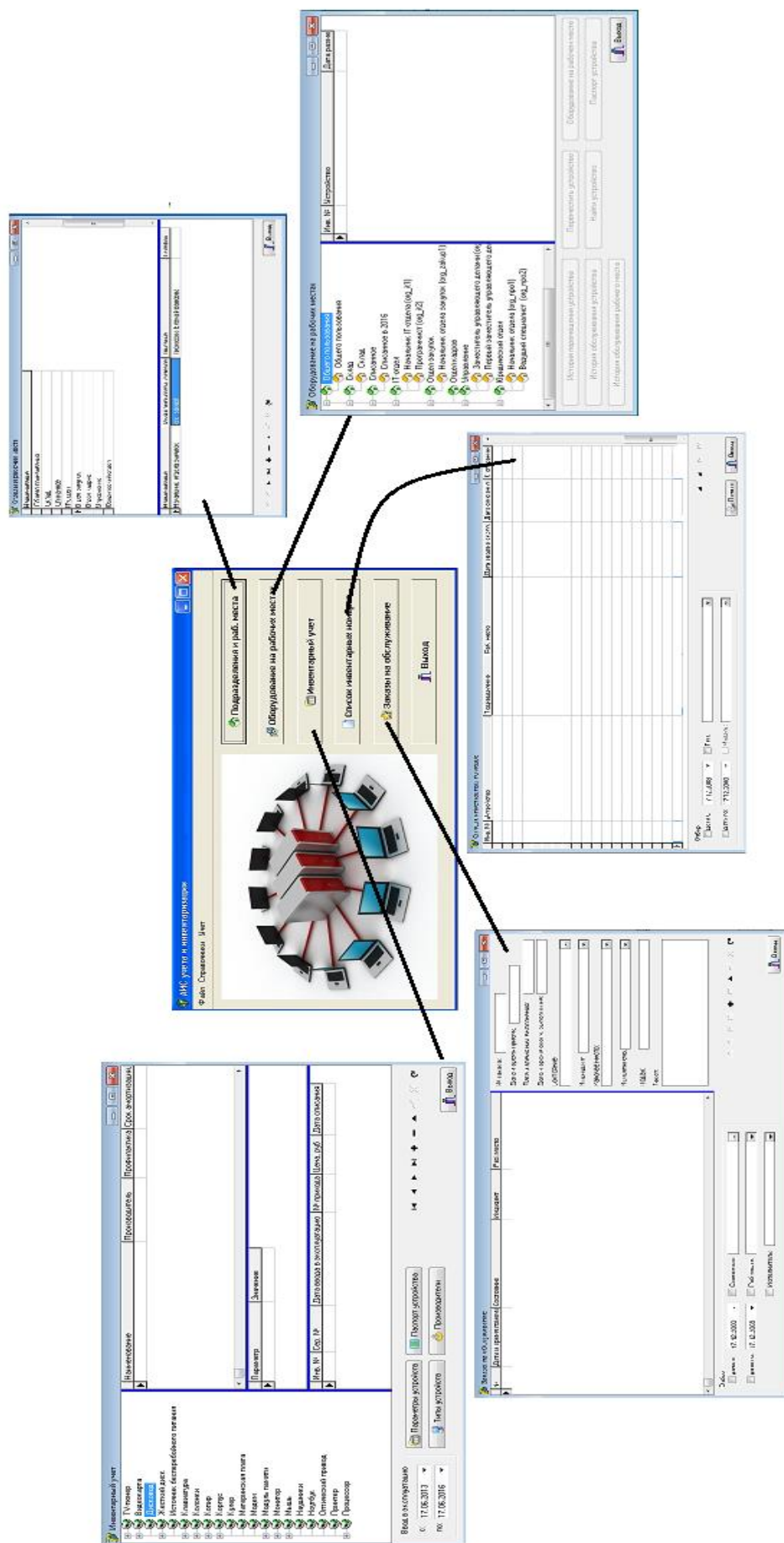


Рисунок 15- Пример интерфейса

3. Выбор и обоснование инструментальных средств разработки и документирования.

Для разработки были выбраны определённые средства и технологии, позволяющие облегчить разработку информационной системы на этапах анализа, проектирования, документирования и других. Выбор каждого из средств обусловлен рядом преимуществ, описанных ниже.

3.1 Выбор операционной системы.

В качестве операционной системы для разработки выбрана операционная система Microsoft Windows 7. Данный выбор основан на том, что Microsoft Windows 7 на данный момент одна из самых распространённых операционных систем, большинство пользователей ПК предпочитают именно Microsoft Windows 7. При сохранении высоких показателей надёжности, безопасности и быстродействия, система является простой и понятной в освоении.

3.2 Выбор средства для документирования.

Для ведения документации по проекту применяется MS Word 2010. Данный инструмент выбран, в первую очередь, по причине популярности среди пользователей ПК. Также, исходя из имеющегося опыта использования данного редактора, который позволяет сократить сроки, требующиеся для разработки программного комплекса, нет необходимости выделять дополнительное время на изучение другого инструмента.

3.3 Выбор средств проектирования

Для разработки функциональной модели выбрано CASE-средство Computer Associates Process Modeller 7.2(BPwin), благодаря описанным ниже его особенностям. BPwin это мощный инструмент для создания моделей, которые позволяют документировать, планировать и анализировать изменения в сложных бизнес-процессах. BPwin является средством для сбора нужной информации о работе отдела, предприятия, организации и графического изображения полученной информации в виде непротиворечивой и целостной модели. В BPwin поддерживается три

методологии: IDEF0, DFD и IDEF3, позволяющие анализировать бизнес с трех точек зрения, являющихся ключевыми:

- с точки зрения функциональности системы. В рамках нотации IDEF0 бизнес-процесс, может быть представлен в виде набора элементов-работ, взаимодействующих между собой, а также показываются производственные, информационные и людские ресурсы, которые потребляются каждой работой;

- с точки зрения документооборота (потоков информации) в исследуемой системе. Диаграммы DFD способны дополнить уже отраженную в модели IDEF3 информацию, так как они являются диаграммами, описывающими потоки данных, и позволяют проследить, каким именно образом происходит обмен информацией между разными бизнес-функциями, находящимися внутри системы. Однако диаграммы DFD оставляют без внимания взаимодействие между бизнес-функциями, и это является отрицательным моментом;

- с точки зрения последовательности выполняемых работ: еще более точную картину можно получить, дополнив модель диаграммами IDEF3. Этот метод привлекает внимание к очередности выполнения событий.

Существуют два уровня представления модели, физический и логический. Логический уровень представления модели, является абстрагированным от конкретной реализации взглядом на обрабатываемые данные. Логическая модель данных это универсальный и никак не связанный с типом конкретной СУБД вариант представления. Физическая же модель данных, напротив, полностью зависит от используемой СУБД, фактически отображая её системный каталог. Физическая модель несет в себе информацию о всех объектах БД. Поскольку стандарты на объекты БД отсутствуют (например, нет стандарта на типы данных), физическая модель очень прочно зависит от используемой реализации СУБД. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей.

Для инфологического проектирования базы данных было выбрано CASE-средство Computer Associates ERwin 7.2, благодаря описанным ниже его особенностям.

Создание модели данных, обычно, начинается с подробного описания логической модели, реализуемой специалистом. После создания проектной логической модели, специалист может спокойно выбрать необходимую СУБД и ERwin в автоматическом режиме создаст соответствующую ей физическую модель. Основываясь на выбранной физической модели, ERwin может сгенерировать системный каталог СУБД или соответствующий SQL-скрипт. Такой процесс носит название прямого проектирования (Forward Engineering). Тем самым достигается отличная масштабируемость – создав всего одну логическую модель данных, вполне можно генерировать физические модели под любую из множества поддерживаемых ERwin СУБД. С другой стороны, ERwin способен по содержимому системного каталога или SQL-скрипту воссоздать и физическую, и логическую модель данных (Reverse Engineering). На основе полученной логической модели данных можно генерировать физическую модель для другой СУБД и затем генерировать ее корневой каталог.

Таким образом, средствами Erwin можно решить задачу по переносу структуры базы данных между серверами.

Существуют следующие уровни физической модели:

- модель СУБД (DBMS Model).
- трансформационная модель (Transformation Model);

Физическая модель несет в себе всю информацию, которая необходима для реализации конкретной БД. Для реализации отдельного проекта, который может быть частью общей ИС и описывать подмножество предметной области, используется трансформационная модель. В ERwin есть поддержка ведения отдельных проектов, так как средства данного пакета позволяют проектировщику выделять подмножество модели в виде предметных областей (Subject Area). Трансформационная модель позволяет

проектировщикам и администраторам БД гораздо лучше и четче представлять, какие именно объекты БД хранятся в словаре данных, и проверять, насколько физическая модель данных удовлетворяет требованиям к ИС.

Модель СУБД автоматически генерируется из трансформационной модели и является точным отображением системного каталога СУБД. ERwin непосредственно поддерживает эту модель путем генерации системного каталога.

3.4 Выбор СУБД. Проектирование серверной части

К СУБД, которая используется для разработки и использования базы данных, логично выдвинуть следующие требования:

- требование надежности (СУБД должна обеспечивать возможность пользователям и системным администраторам восстановления предыдущего состояния СУБД без критичной потери данных);
- требование защиты информации (в составе СУБД должны быть средства идентификации и аутентификации, должна предоставляться возможность разграничения доступа к различным объектам базы данных – разные группы пользователей должны иметь разные права на доступ к объектам базы данных);
- требование модифицируемости (база данных должна быть легко способна к расширению путем добавление необходимых объектов);
- требование минимизации затрат на сопровождение и поддержку;
- требование эргономичности.

Перечислим основные характеристики СУБД и рассмотрим их в отношении основных СУБД:

- 1) размер базы данных:
 - до сотни мегабайт: MS Access, Paradox, Dbase, Foxpro/VFP;
 - гигабайты: MySQL, PostgreSQL, Interbase, Informix;
 - сотни гигабайт и больше: MS SQL Server, Oracle, SyBase, DB/2.

- 2) количество одновременных пользователей:
 - до десятка пользователей: Paradox, Dbase, Foxpro/VFP, MS Access;
 - десятки пользователей: MySQL, Informix;
 - сотни пользователей: Interbase, PostgreSQL;
 - тысячи пользователей: MS SQL Server, Oracle, SyBase, DB/2;
- 3) платформа:
 - Windows: MS SQL Server, SyBase, Paradox, Dbase, Foxpro/VFP, MS Access;
 - Windows+Linux: Oracle, MySQL, Interbase, DB/2;
- 4) язык программирования:
 - языки от Microsoft: Foxpro/VFP, SyBase, Foxpro/VFP, MS Access, MS SQL Server;
 - языки от Borland: MS SQL Server, Interbase, Paradox, MS Access;
- 5) защита данных:
 - слабая: Paradox, Dbase, Foxpro/VFP, MS Access;
 - сильная: MS SQL Server, Oracle, SyBase, DB/2, Interbase, MySQL, PostgreSQL, Paradox;
- 6) требования к техническому оснащению:
 - неприхотливые: PostgreSQL, MySQL, Paradox, Dbase, Foxpro/VFP, MS Access;
 - чувствительные: Interbase, Informix, SyBase;
 - требуют отдельных мощных серверов с большой RAM, желательно на нескольких процессорах: MS SQL Server, Oracle, DB/2;

Основываясь на перечисленных выше критериях выбора СУБД был сделан выбор в пользу MS SQL Server 2014.

В разрабатываемом программном обеспечении предполагается создание процедур обработки данных корпоративной автоматизированной информационной системе, предназначенной для размещения на

автоматизированном рабочем месте пользователя. Список предполагаемых процедур:

- 1) процедуры, необходимые для извлечения данных из базы данных, в соответствии с требованиями проанализированной предметной области (данные таблиц, сведения из определенных запросов к таблицам и т.п.);
- 2) процедуры, которые поддерживают редактирование данных и корректную запись их в базу данных;
- 3) процедуры, создаваемые для удаления данных из базы в соответствии с запросами пользователя;
- 4) процедуры, обеспечивающие целостность базы данных;
- 5) процедуры, обеспечивающие защиту базы данных от несанкционированного доступа;
- 6) процедуры, обеспечивающие контроль вводимой оператором информации, созданные для уменьшения процента ошибок пользователя при вводе данных;

3.5 Выбор и обоснование языка программирования

Для написания клиентского приложения выбор был сделан в пользу CodeGear Delphi 2007. При выборе среды программирования были рассмотрены три альтернативы: Microsoft Visual C#, Delphi 7 и C++ Builder 6.0. Среда Delphi использует язык ObjectPascal, который предоставляет следующие преимущества по сравнению с аналогичными программными продуктами:

- высокая скорость разработки приложения (RAD);
- отличная производительность разработанного приложения;
- небольшие требования разработанного ПО к ресурсам компьютера;
- наращиваемость за счет подключения дополнительных компонент и инструментов в среду Delphi;
- возможность разработки новых компонент и блоков собственными средствами Delphi (существующие компоненты и инструменты доступны в исходных кодах);
- удачная проработка иерархии объектов.

4. Программная реализация автоматизированной информационной системы.

При построения системы использовалась двухзвенная архитектура клиент-сервер, которая предполагает разделение приложения на серверную и клиентскую части. Серверная часть включает в себя базу данных и отвечает за управление данными, администрирование, обеспечение защиты и разделение информации. Клиентская часть, включающая в себя персональные программы, обеспечивает графический интерфейс для работы с БД и размещается на компьютерах пользователей.

Достоинствами клиент-серверной архитектуры являются:

- эффективность работы приложения за счет использования мощности сервера;
- малая загрузка сети;
- обеспечение хорошего контроля целостности данных;
- большие возможности администрирования базы данных.

К недостаткам данной архитектуры можно отнести сильную загруженность сервера.

2.3.1 Функциональные модули системы

Для разработки программного продукта был использован модульный принцип программирования. По данному методу программная реализация включает множество процедур и функций некоторого числа форм и модулей данных. Такая организация программного обеспечения позволяет легко отслеживать ошибки.

В соответствии с разработанной структурой приложения, следует выделить следующие функциональные модули:

- модуль главное меню;
- модуль работы со справочниками;
- модуль формирования входной информации;
- модуль формирования документов;

- модуль учета оборудования;
- модуль учета рабочих мест;
- модуль заказов на обслуживание;
- модуль инвентарного учета.

На рисунке 16 в виде схемы представлены функциональные модули системы.

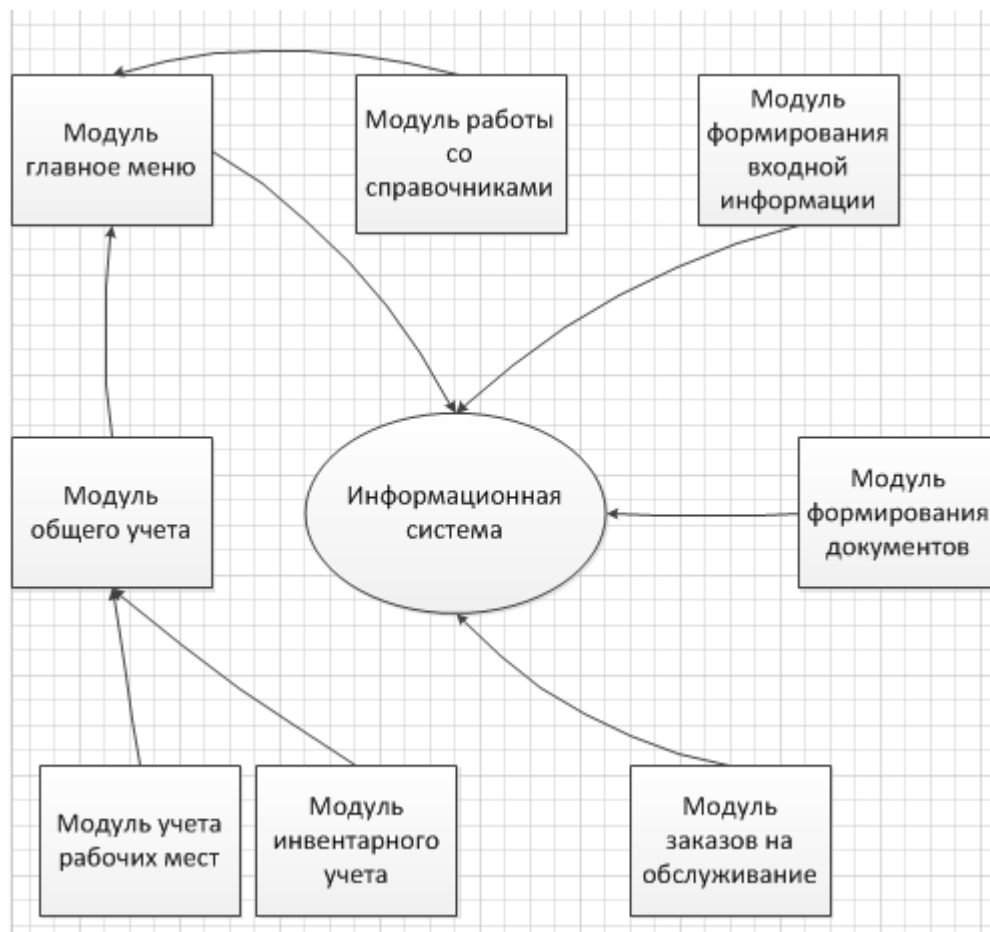


Рисунок 16 - Функциональные модули системы

Модуль «Главное меню» используется для запуска основных процедур информационной системы и для завершения работы с программой.

Модуль работы со справочной информацией включает в себя следующие справочники:

- типы устройств;
- параметры устройств;
- сотрудники IT-отдела;

- поставщики;
- производителя;
- состояния заказа.

Назначением данного модуля является поиск и предоставление пользователю справочной информации.

Модуль «Формирование входной информации» используется для ввода первичных данных и просмотра ранее занесенных. Этот модуль выполняет задачи ввода заявок по обслуживанию и ремонту компьютерной техники, а также формированию номенклатуры компьютерной техники с помощью разработанных форм.

Модуль «Формирование документов» используется для формирования печатных форм. Отчеты формируются, с использованием запросов, которые обрабатывают исходную информацию в соответствии с заданными параметрами пользователя.

Модули учета используются для обработки данных находящихся в системе, а также для учета рабочих мест и оборудования.

Алгоритмы работы программы являются типовыми алгоритмами работы с базой данных. В основном все алгоритмы работы связаны с вводом данных от пользователя, проверке введенной информации на предмет нарушения целостности данных и занесение введенной информации в саму базу, если введенные сведения не нарушают целостности. Пример кода формы “Инвентарный учет”:

```
unit uInventory;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DBCtrls, StdCtrls, ExtCtrls, Grids, DBGrids, ComCtrls, Buttons,
  ImgList;
type
  TfmInventory = class(TForm)
    Panel2: TPanel;
    Panel3: TPanel;
    DBGrid2: TDBGrid;
    DBGrid3: TDBGrid;
```

```

DBNavigator1: TDBNavigator;
bbitExit: TBitBtn;
GroupBox1: TGroupBox;
Label3: TLabel;
Label4: TLabel;
dtpDate1: TDateTimePicker;
dtpDate2: TDateTimePicker;
bbitTypes: TBitBtn;
bbitParam: TBitBtn;
TreeView1: TTreeView;
DBGrid1: TDBGrid;
Splitter2: TSplitter;
ImageList1: TImageList;
Splitter1: TSplitter;
Splitter3: TSplitter;
bbitDevicePassport: TBitBtn;
BitBtn1: TBitBtn;
procedure FormShow(Sender: TObject);
procedure DBGrid1Enter(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure dtpDate1Change(Sender: TObject);
procedure bbitPassportClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure TreeView1Change(Sender: TObject; Node: TTreeNode);
procedure bbitDevicePassportClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    fmInventory: TfmInventory;
implementation
uses uDM, DB, DateUtils, uMain, uRep, uTypes;
{$R *.dfm}
procedure TfmInventory.FormShow(Sender: TObject);
    //Добавление вершин - комплектующих:
    procedure AddChild(Node: TTreeNode);
    var
        Child: TTreeNode;
    begin
        DM.dstModels.Open;
        DM.dstModels.First;
        while not DM.dstModels.Eof do
            begin
                //Добавление вершины - комплектующего:

```

```

    Child := TreeView1.Items.AddChild(Node, DM.dstModels.FieldByName('Наименование').AsString);
    //Data указывает на ID комплектующего:
    Child.Data := Pointer(DM.dstModels.FieldByName('ID').AsInteger);
    Child.ImageIndex := 1;
    DM.dstModels.Next;
end;
end;
var
    CategTreeNode: TTreeNode;
begin
    TreeView1.Items.Clear;
    DM.dstTypes.Open;
    DM.dstTypes.Sort := 'Наименование';
    DM.dstDevices.Open;
    DM.dstIncome.Open;
    DM.dstModels.Open;
    DM.dstModelPar.Open;
    DM.dstParameters.Open;
    DM.dstProducers.Open;
    dtpDate1.Change(Self);
    DM.dstTypes.First;
    while not DM.dstTypes.Eof do
    begin
        //Добавление вершины - подразделения:
        CategTreeNode := TreeView1.Items.Add(nil, DM.dstTypes.FieldByName('Наименование').AsString);
        //Data указывает на ID подразделения:
        CategTreeNode.Data := Pointer(DM.dstTypes.FieldByName('ID').AsInteger);
        CategTreeNode.ImageIndex := 0;
        AddChild(CategTreeNode); //Добавление списка рабочих мест по текущему подразделению
        DM.dstTypes.Next;
    end;
end;
//DBNavigator1 привязывается к выбранному DBGrid'у и соответствующему набору данных:
procedure TfmInventory.DBGrid1Enter(Sender: TObject);
begin
    DBNavigator1.DataSource := (Sender as TDBGrid).DataSource;
end;
procedure TfmInventory.FormCreate(Sender: TObject);
begin
    dtpDate1.Date := IncYear(Date, -3);
    dtpDate2.Date := Date;
end;
procedure TfmInventory.dtpDate1Change(Sender: TObject);
begin
    DM.dstDevices.Close;
    //По диапазону дат ввода в эксплуатацию:

```



```

DM.dstDevices.Parameters.ParamByName('pDate1').Value := dtpDate1.Date;
DM.dstDevices.Parameters.ParamByName('pDate2').Value := dtpDate2.Date;
DM.dstDevices.Open;
end;

procedure TfmInventory.bbtPassportClick(Sender: TObject);
begin
    if DM.dstDevices.RecordCount <=0 then
        Application.MessageBox('Не выбрано устройство!', 'Ошибка', MB_OK+MB_ICONERROR)
    else
        Rep_Passport(DM.dstDevices.FieldByName('ИнвN').AsInteger);
    end;
end;

procedure TfmInventory.FormClose(Sender: TObject;
    var Action: TCloseAction);
begin
    DM.dstModelPar.Close;
    DM.dstModels.Close;

    //Набор данных остается открытым, если открыты формы, где он используется:
    if not fmType.Showing then
        DM.dstTypes.Close;
        DM.dstParameters.Close;
        DM.dstProducers.Close;
    end;

    //При выборе на дереве элемента в таблице выполняется переход на соответствующие
    //комплектующее или тип комплектующих:
    procedure TfmInventory.TreeView1Change(Sender: TObject; Node: TTreeNode);
    begin
        if DM.dstTypes.Active and DM.dstModels.Active then
            if Assigned(TreeView1.Selected) then
                if not Assigned(TreeView1.Selected.Parent) then
                    //если выбран тип комплектующих:
                    DM.dstTypes.Locate('ID', Integer(Node.Data), [])
                else
                    begin
                        //если выбрана модель:
                        DM.dstTypes.Locate('ID', Integer(Node.Parent.Data), []);
                        DM.dstModels.Locate('ID', Integer(Node.Data), []);
                    end;
                end;
            end;
        end;

    //Формирование паспорта устройства:
    procedure TfmInventory.bbtDevicePassportClick(Sender: TObject);
    begin
        Rep_Passport(DM.dstDevices.FieldbyName('ИнвN').AsInteger);
    end;
end.

```

Заключение

В ходе выполнения дипломного проекта были изучены принципы учета и инвентаризации оборудования, а также приема и выполнения заказов на обслуживание и ремонт. В ходе работы были разработаны, созданы и отлажены все компоненты системы и проведена следующая работа:

1. Проведен анализ предметной области, который помог определить требования пользователей, выявить функциональные и нефункциональные требования к разрабатываемой информационной системе; анализ предметной области был проведен с использованием среды функционального моделирования BPWin.
2. По результатам анализа предметной области была получена инфологическая модель и спроектирована даталогическая модель базы данных применительно к СУБД MS SQL Server 2014; проектирование базы данных выполнялось в среде ERWin.
3. Разработана структура информационной системы и определены основные функциональные блоки.
4. Спроектирован лаконичный и дружелюбный интерфейс пользователя, позволяющий пользователю работать в среде системы.
5. Проведена программная реализация автоматизированной системы отдела в среде разработки Code Gear Rad Studio Delphi 2007.
6. Проведена проверка работоспособности системы на модельной базе данных.
7. Разработанная система после незначительной доработки может использоваться в любой организации для учета и инвентаризации технических и программных средств организации, а также приема и обработки заказов на обслуживание и ремонт оборудования.

5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Иванова Г.С. Технология программирование, - М.: МГТУ им. Н.Э. Баумана, 2006г. – 336 с.
2. ИНТУИТ. Национальный открытый университет. Лекция 6: Методологии моделирования предметной области. [Электронный ресурс]. URL: <http://www.intuit.ru/studies/courses/2195/55/lecture/1628> (дата обращения 10.01.14).
3. ИНТУИТ. Национальный открытый университет. Лекция 4: Анализ предметной области. [Электронный ресурс]. URL: <http://www.intuit.ru/studies/courses/574/430/lecture/9749>.
4. Маклаков С.В. Создание информационных систем с AllFusionModelingSuite, М.: ДИАЛОГ МИФИ, 2003г. – 432 с.
5. Фаулер М. UML. Основы / 3-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2004. – 192 с.
6. Крис Дж. Дейт. Введение в системы баз данных, - Вильямс, 2006г. – 1328 с.
7. Верхолат А.М., Гаврилов В.А. Проектирование структуры базы данных, БГТУ 2007г.
8. Гулятьева А.К., Машина В.А.. Проектирование и дизайн пользовательского интерфейса– Издательство: Захаров, 2007г.
9. Тернстрем Т., Вебер Э., Хотек М. SQLServer 2008. Разработка баз данных. Москва 2012г.
10. Карли Уотсон, Кристиан Нейгел, Якоб Хаммер Педерсен, Джон Д. Рид, Морган Скиннер, Эрик Уайт. DELPHI 2007: базовый курс / Пер. с англ. – М.: ООО «И.Д. Вильямс», 2009г. – 1216 с.
11. Орлов С.А. «Технология разработки программного обеспечения», - СПб, Питер, 2003г. – 56 с.
12. Фленов М.Е. Библия DELPHI – СПб.: БХВ –Петербург, 2009г. – 560 с.

Приложение А(Программные формы)

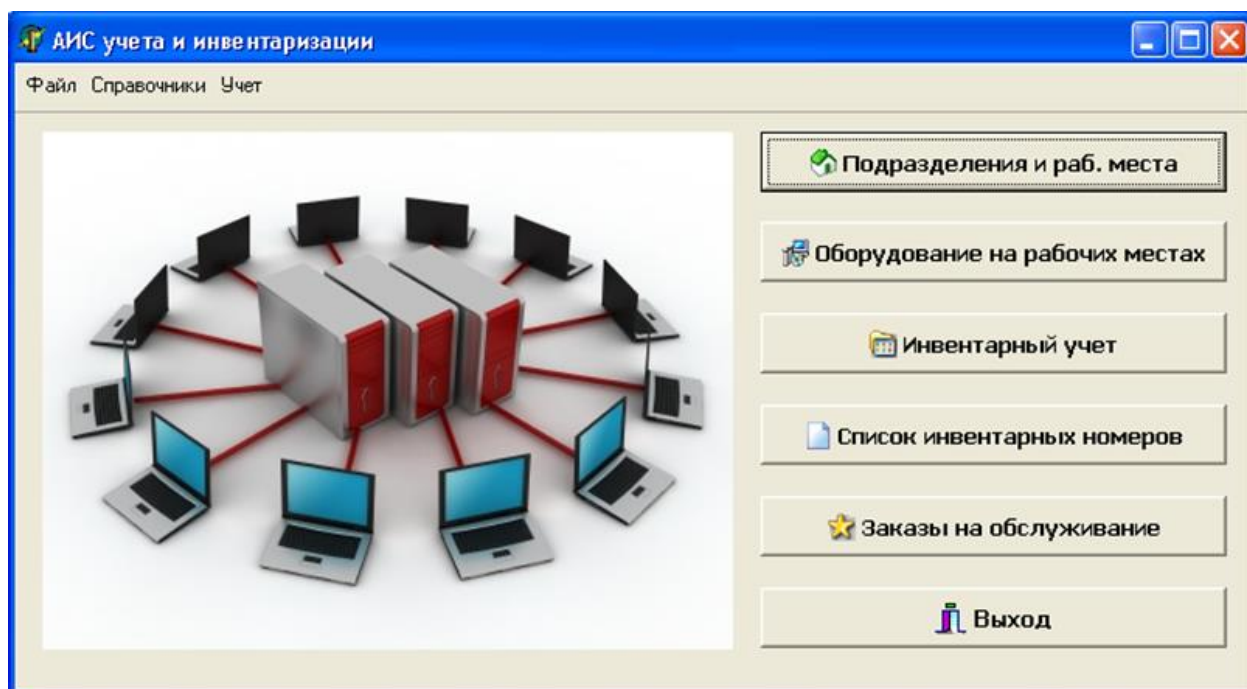


Рисунок А.1 – Главная форма

№	Дата и время приема	Состояние	Инцидент	Раб. место
0021	18.06.16 12:12:12	Оборудование ожидает вывоза	Сбой	Начальник отдела закупок
0019	18.06.16 16:12:12	Передан на исполнение	Сбой	Ведущий специалист

№ заказа: 0021
 Дата и время приема: 18.06.16 12:12:12
 Дата и время нач. выполнения:
 Дата и время оконч. выполнения:
 Состояние: Оборудование ожидает вывоза в ремонт
 Инцидент: Сбой
 Рабочее место: Начальник отдела закупок
 Исполнитель: Макаренко И.Д.
 Подал: Никитин
 Текст: Компьютер не включается

Отбор
☒ дата с: 17.06.2016 ☐ Состояние:
☒ дата по: 19.06.2016 ☐ Раб место:
☐ Исполнитель:

Рисунок А.2 – Форма учета заказов

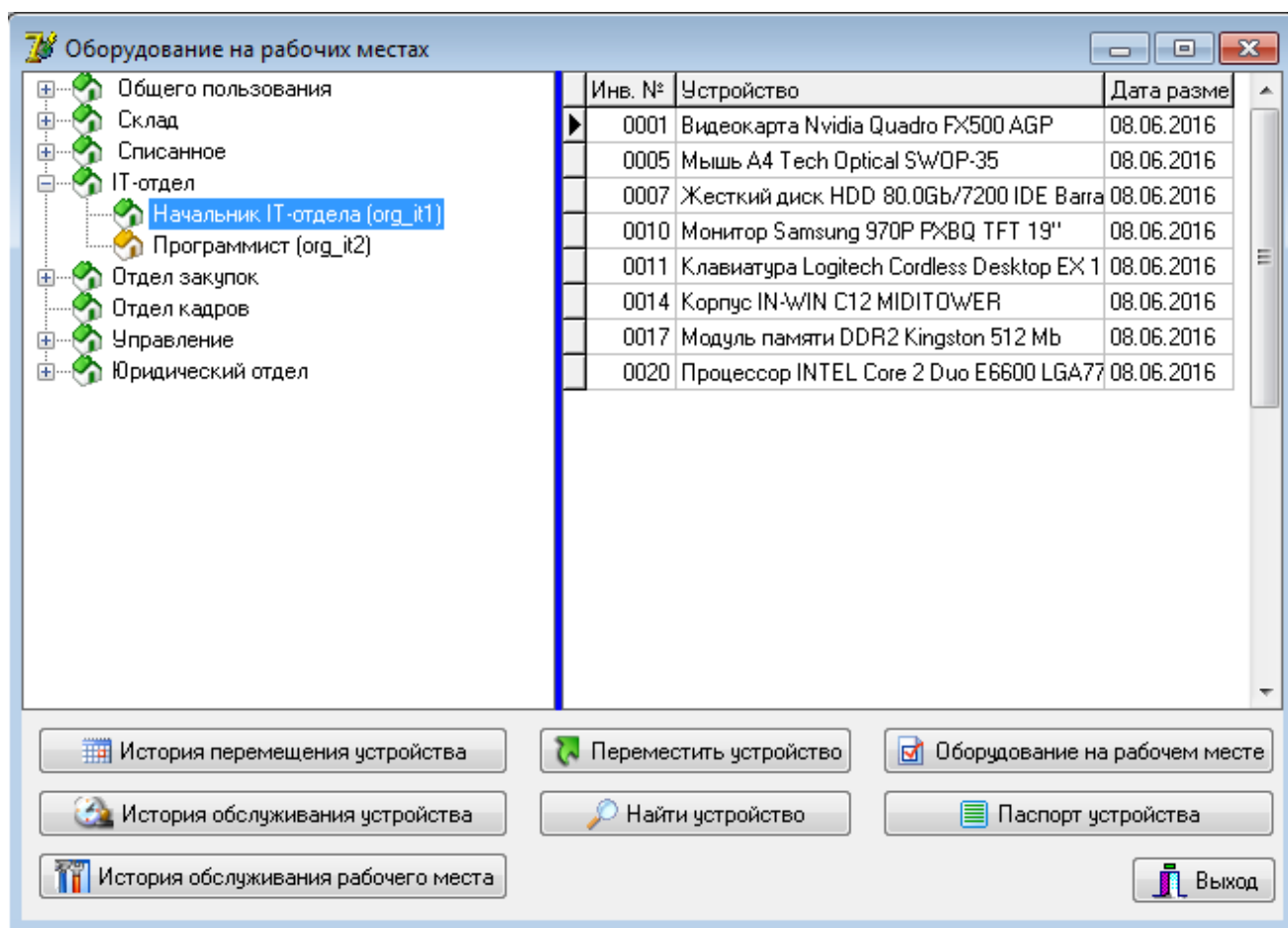


Рисунок А.3 – Оборудование на рабочих местах

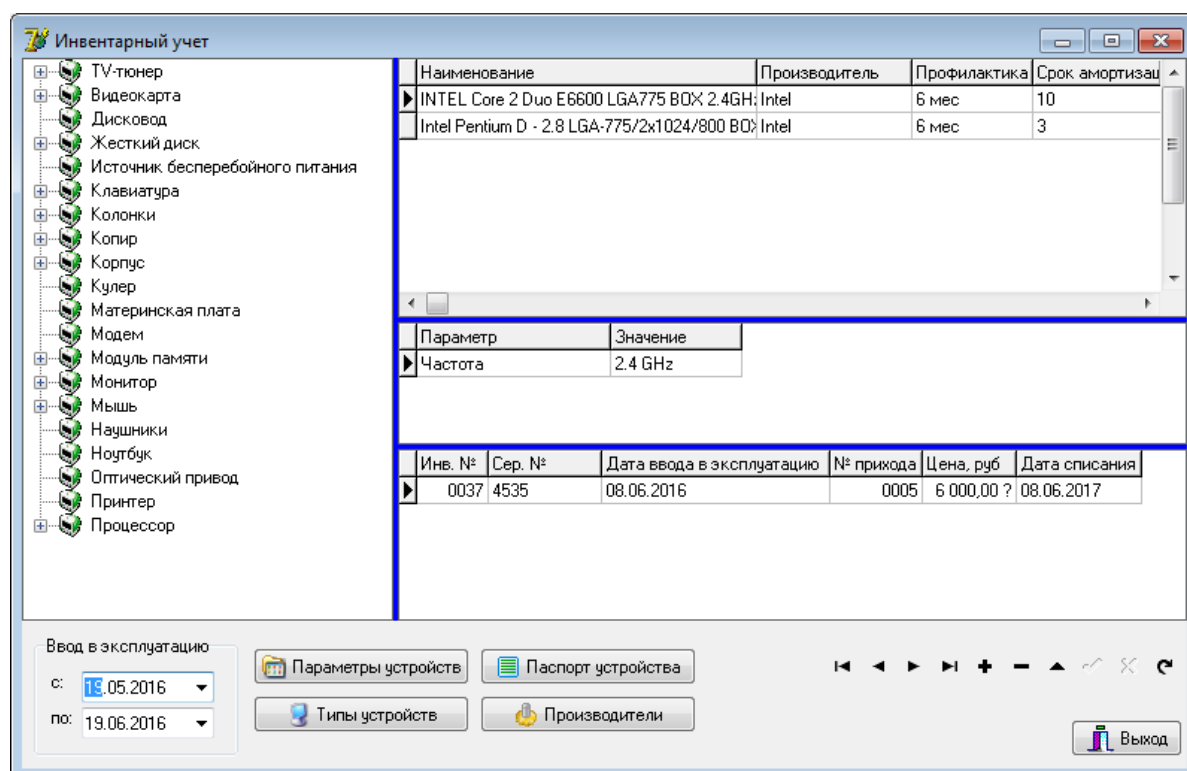


Рисунок А.4 – Инвентарный учет

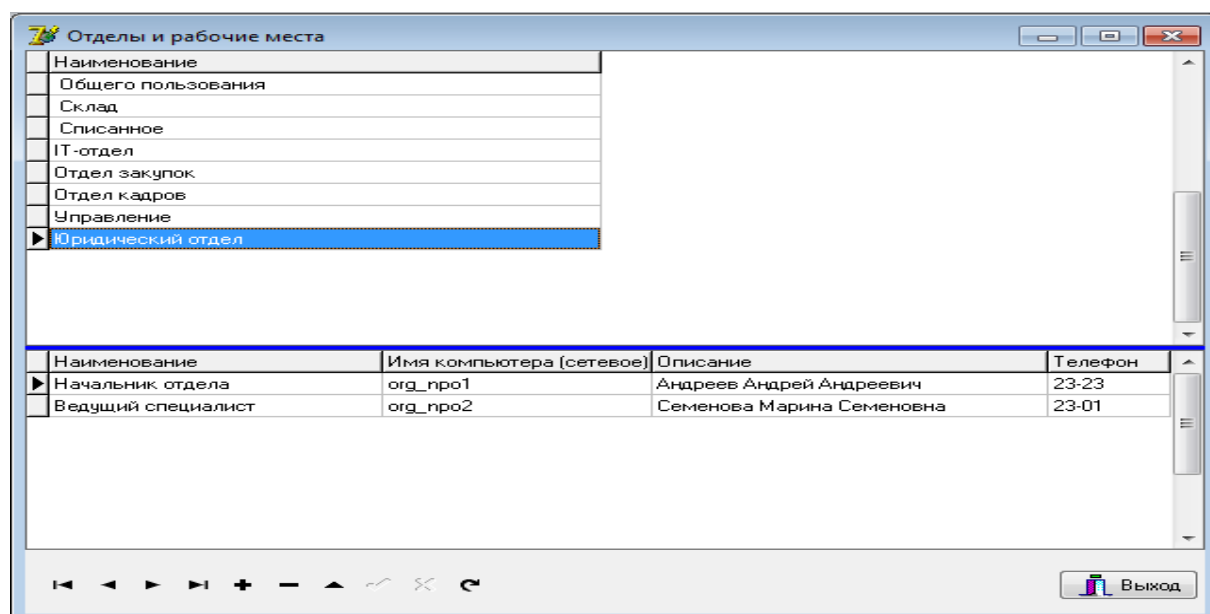


Рисунок А.5 – Список отделов и рабочих мест

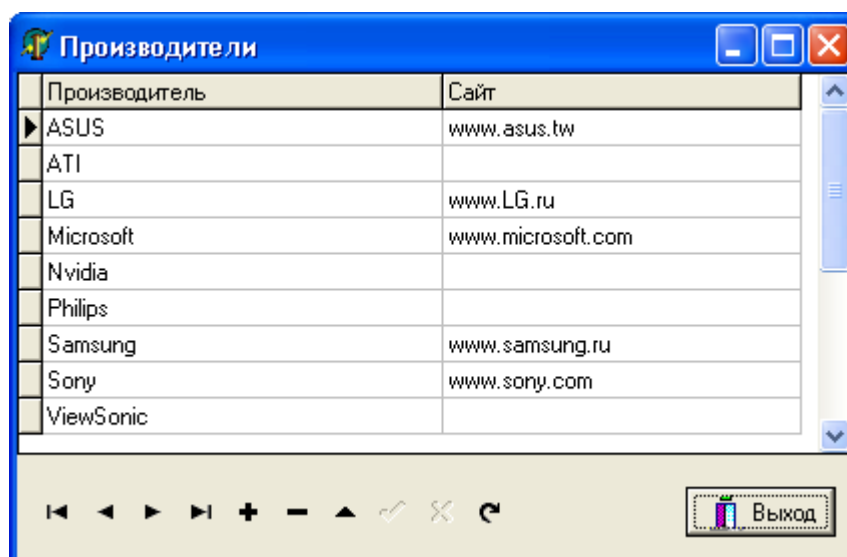


Рисунок А.6 – Справочник производителей

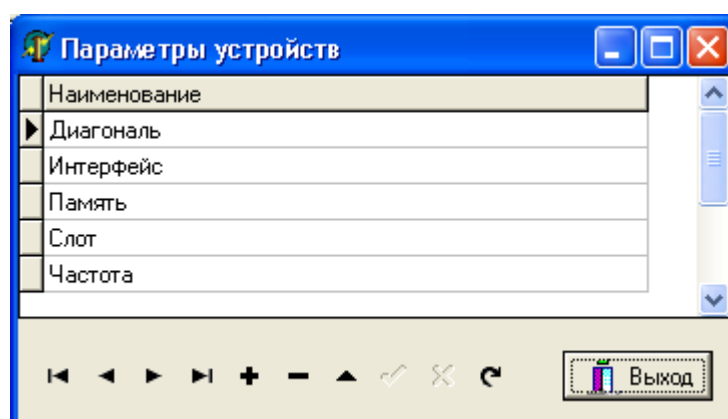


Рисунок А.7 – Справочник параметров оборудования

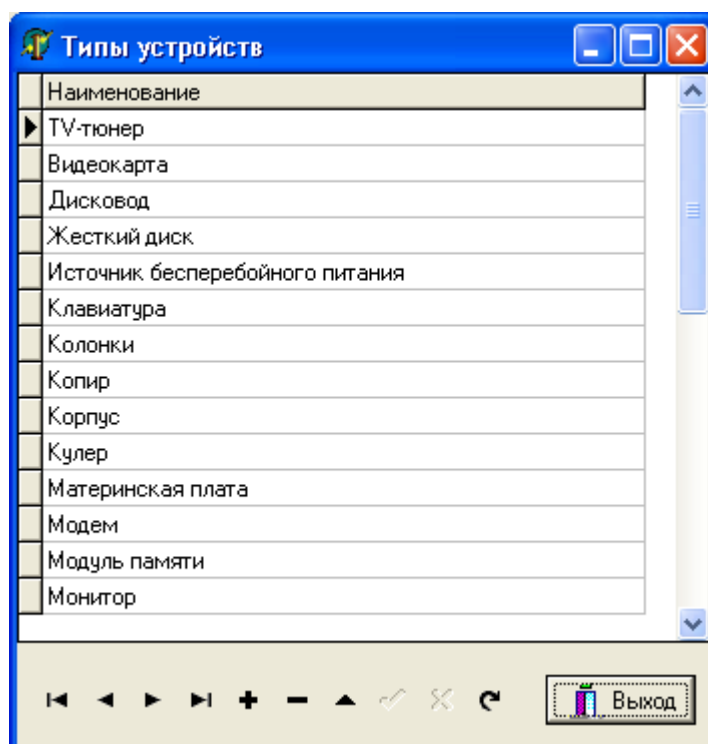


Рисунок А.8 – Справочник типов устройств

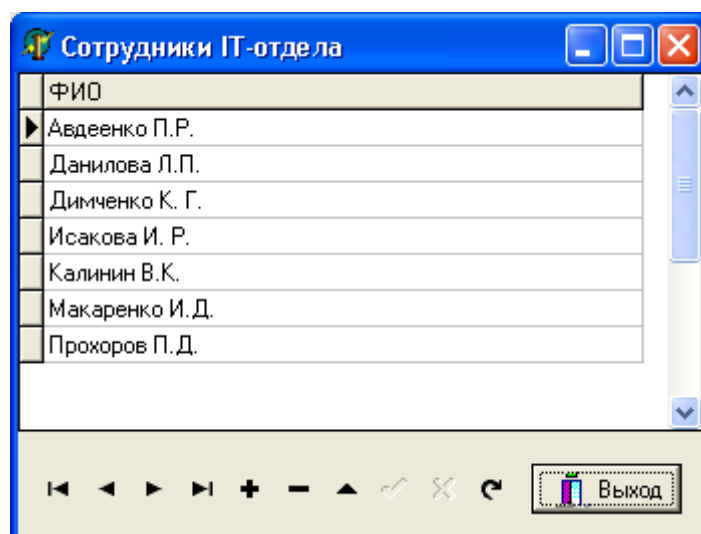


Рисунок А.9 – Справочник работников

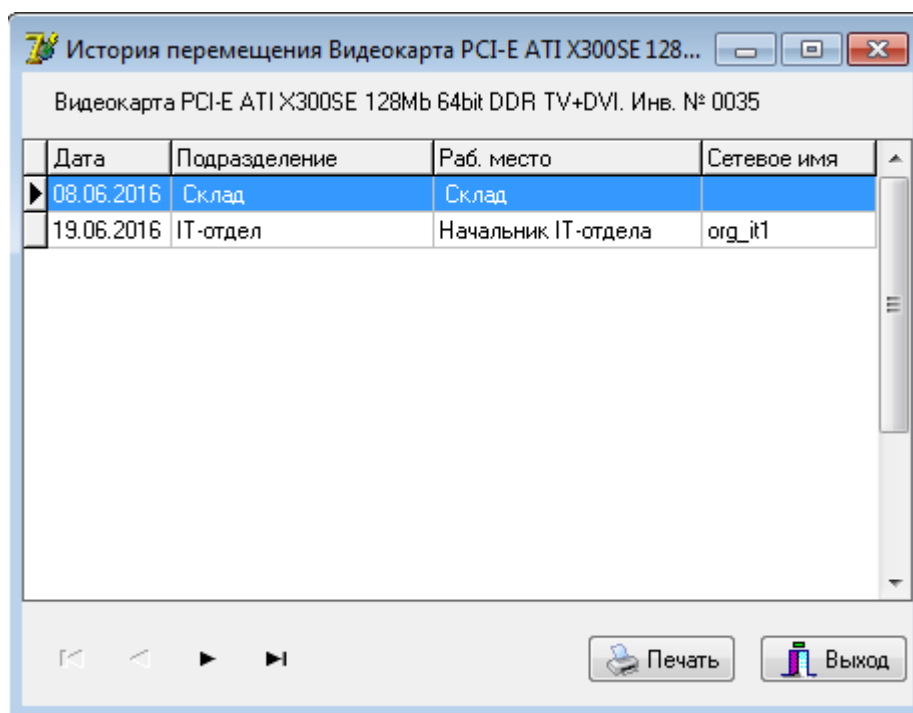


Рисунок А.10 – История перемещения устройства

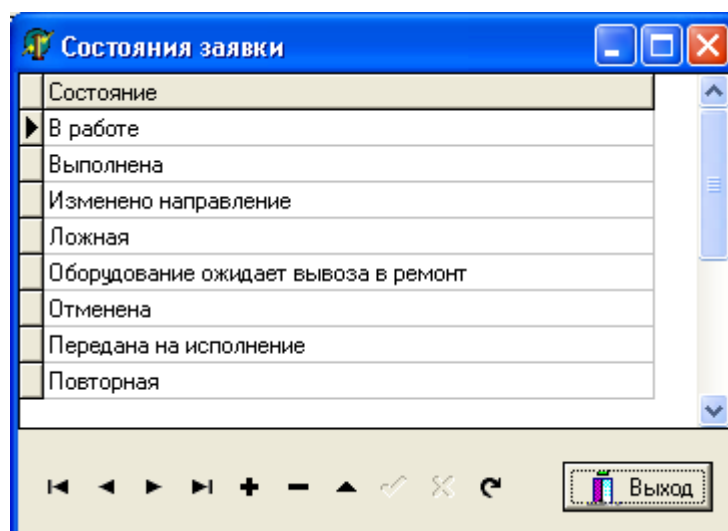


Рисунок А.11 – Справочник состояний заявки

Список инвентарных номеров

Инв. №	Устройство	Подразделение	Раб. место	Дата ввода в экпл.	Дата списания	К списанию
0029	Монитор ViewSonic LCD 19" VA903M	Склад	Склад	20.03.2009	20.03.2012	Да
0030	Видеокарта ATI RADEON HD5400	IT-отдел	Программист	20.05.2016		Да
0031	Модуль памяти DDR3 Kingston 4 Gb	IT-отдел	Программист	08.06.2016	08.06.2019	Нет
0032	Видеокарта Nvidia Quadro FX500 AGP	Склад	Склад	08.06.2016	08.06.2019	Нет
0033	Процессор Intel Pentium D - 2.8 LGA-775/2	Склад	Склад	08.06.2016	02.07.2026	Нет
0034	Мышь A4 Tech Optical SW/DP-35	Склад	Склад	08.06.2016	08.06.2017	Нет
0035	Видеокарта PCI-E ATI X300SE 128Mb 64b	IT-отдел	Начальник IT-отдела	08.06.2016	08.06.2018	Нет
0036	Видеокарта PCI-E ATI X1550 256Mb 128b	Склад	Склад	08.06.2014	08.06.2016	Да
0037	Процессор INTEL Core 2 Duo E6600 LGA7	Склад	Склад	08.06.2016	08.06.2017	Нет
0038	Монитор Samsung SyncMaster 753 DFX	Склад	Склад	20.05.2012	20.06.2017	Нет
0039	Монитор ViewSonic LCD 19" VA903M	Склад	Склад	20.05.2008	20.05.2017	Нет
0040	Монитор ViewSonic LCD 17" VA703B	Склад	Склад	20.04.2010	20.04.2017	Нет
0041	Монитор Samsung 970P PxBQ TFT 19"	Склад	Склад	21.03.2012	20.03.2017	Нет
0042	Клавиатура Logitech Cordless Desktop EX	Склад	Склад	20.06.2015	20.06.2018	Нет
0043	Клавиатура Logitech Cordless Desktop MX	Склад	Склад	20.06.2015	20.06.2018	Нет
0044	Клавиатура Logitech Cordless Destop MX	Склад	Склад	20.05.2015	20.06.2017	Нет
0045	Колонки Genius SP-K10	Склад	Склад	20.06.2015	20.06.2017	Нет
0046	Копир Xerox Laser TN-12	Склад	Склад	20.06.2015	20.06.2018	Нет
0047	Корпус IN-WIN C12 MIDITOWER	Склад	Склад	20.05.2015	20.05.2018	Нет

Отбор
☐ дата с: 01.05.2016 ☐ Тип:
☐ дата по: 18.06.2016 ☐ Модель:

Печать Выход

Рисунок А.12 – Список инвентарных номеров

Список рабочих мест

Наименование	Подразделение	Сетевое имя	Домен/раб. группа	ip-адрес	Программное обеспечение
Общего пользования					
Склад					
Ведущий специалист		org_npo2	WORK	192.168.0.9	Windows 7, MS Office 2010
Заместитель управляющего д		org_zam	WORK	192.168.0.3	Windows 7, MS Office 2013 Stan
Начальник IT-отдела		org_it1	WORK	192.168.0.6	Windows XP, MS Office 2013 Pro
Начальник отдела		org_npo1	WORK	192.168.0.10	Windows XP, MS Office 2013 Sta
Начальник отдела закупок		org_zakup1	WORK	192.168.0.12	Windows Vista, MS Office 2010
Первый заместитель управля		org_zam1	WORK	192.168.0.2	Windows XP, MS Office 2003 Sta
Программист		org_it2	WORK	192.168.0.15	Windows 7, MS Office 2003 Profe
Списанное в 2016					

Печать Выход

Рисунок А.13 – Список рабочих мест

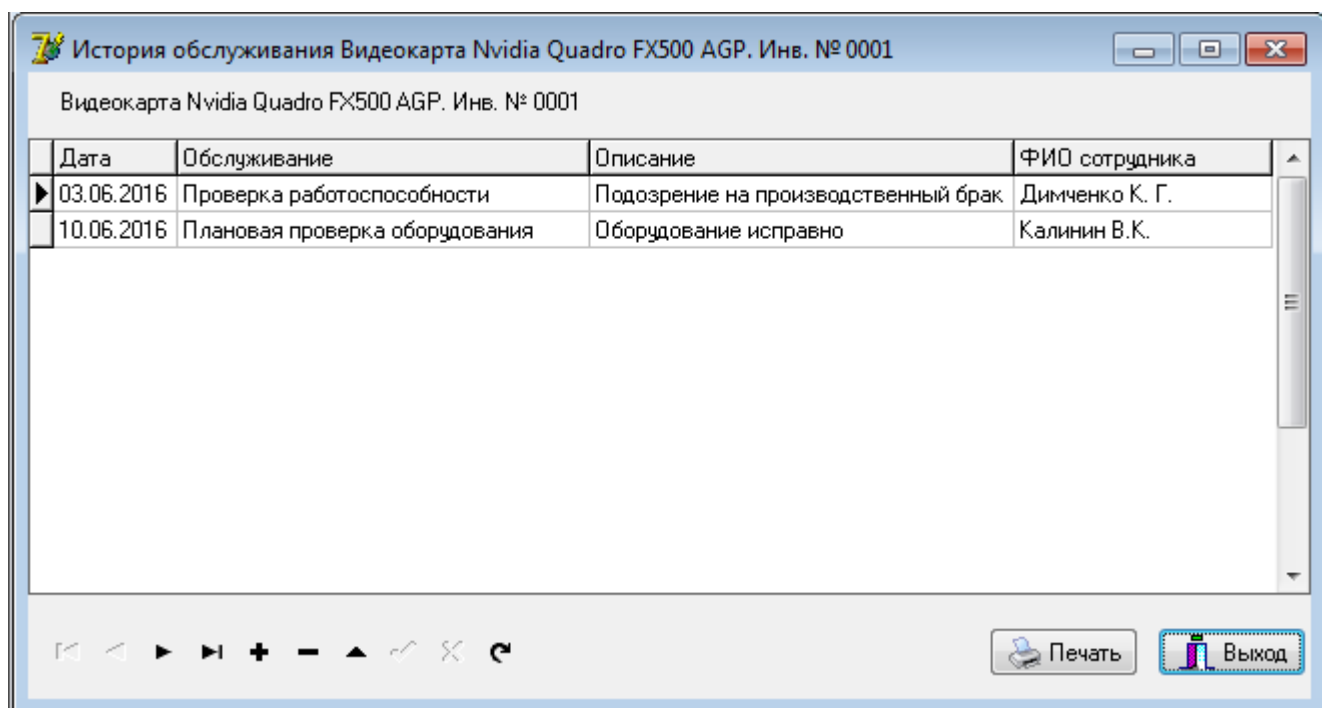


Рисунок А.14 – История обслуживания устройства

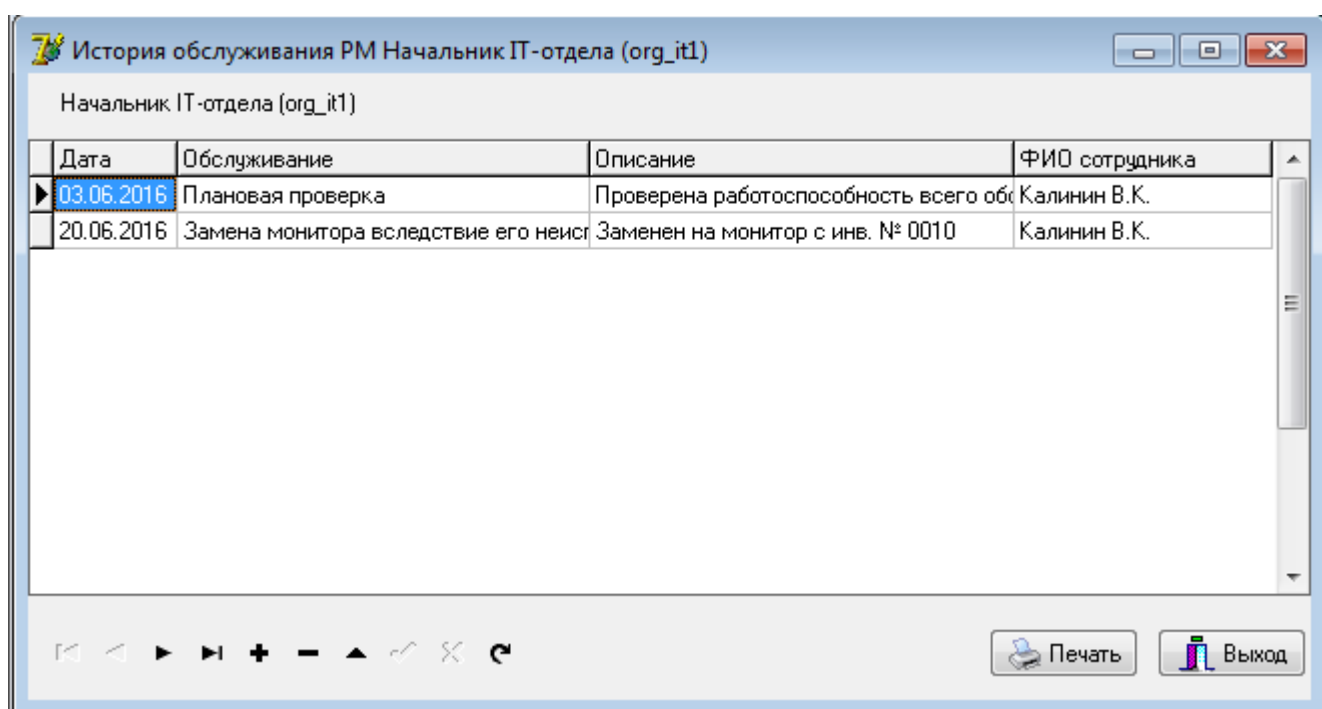


Рисунок А.15 – История обслуживания рабочего места

Рисунок Б.1 – Паспорт устройства

Рисунок Б.2 – Список оборудования на рабочем месте

Общего пользования			
	Инв. №	Устройство	Дата размещения
	8	Копир Xerox Laser TN-12	22.12.2010
	22	Копир Xerox Laser TN-12	22.12.2012
Склад			
	Инв. №	Устройство	Дата размещения
	3	Видеокарта Nvidia Quadro FX500 AGP	08.03.2008
	6	Мышь A4 Tech Optical SWOP-35	15.08.2012
	13	Корпус IN-WIN C12 MIDITOWER	16.08.2012
	16	Модуль памяти DDR2 Kingston 512 Mb	17.08.2012
	18	Процессор INTEL Core 2 Duo E6600 LGA775 BOX 2.4GHz	18.08.2012
	19	Процессор INTEL Core 2 Duo E6600 LGA775 BOX 2.4GHz	19.08.2012
	24	Монитор Samsung 970P PXBQ TFT 19"	20.08.2012
	25	Монитор Samsung SyncMaster 753 DFX	21.08.2012
	26	Видеокарта Nvidia Quadro FX500 AGP	22.08.2012
	28	Колонки Genius SP-K10	23.08.2012
	29	Монитор ViewSonic LCD 19" VA903M	24.08.2012
Ведущий специалист (org_npo2)			
	Инв. №	Устройство	Дата размещения
	2	Видеокарта Nvidia Quadro FX500 AGP	16.12.2008
	9	Монитор Samsung 970P PXBQ TFT 19"	16.12.2009
	12	Клавиатура Logitech Cordless Desktop EX 110	16.12.2010
	21	Процессор INTEL Core 2 Duo E6600 LGA775 BOX 2.4GHz	16.12.2011
Начальник IT-отдела (org_it1)			
	Инв. №	Устройство	Дата размещения
	1	Видеокарта Nvidia Quadro FX500 AGP	08.06.2016 14:30:10
	5	Мышь A4 Tech Optical SWOP-35	08.06.2016 14:30:11
	7	Жесткий диск HDD 80.0Gb/7200 IDE Barracuda V/VII/	08.06.2016 14:29:57

Рисунок Б.3 – Список устройств по АРМам

08.06.2016						
Список рабочих мест						
Наименование	Подразделение	Сетевое имя	Домен/раб. группа	ip-адрес	Программное обеспечение	Описание
Общего пользования	Общего пользования					
Склад	Склад					
Ведущий специалист	Юридический отдел	org_npo2	WORK	192.168.0.9	Windows 7, MS Office 2010	Семенова Марина Васильевна
Заместитель управляющего дела	Управление	org_zam	WORK	192.168.0.3	Windows 7, MS Office 2013 Standart	Макеев Игорь Анатольевич
Начальник IT-отдела	IT-отдел	org_it1	WORK	192.168.0.6	Windows XP, MS Office 2013 Professional	Калинин Василий Кузьмич
Начальник отдела	Юридический отдел	org_npo1	WORK	192.168.0.10	Windows XP, MS Office 2013 Standart	Андреев Андрей Петрович
Начальник отдела закупок	Отдел закупок	org_zakup1	WORK	192.168.0.12	Windows Vista, MS Office 2010	Прохорова Елена Ивановна
Первый заместитель управляющего	Управление	org_zam1	WORK	192.168.0.2	Windows XP, MS Office 2003 Standart	
Программист	IT-отдел	org_it2	WORK	192.168.0.15	Windows 7, MS Office 2003 Professional	

Рисунок Б.4 – Список рабочих мест

Дата	Подразделение	Раб. место	Сетевое имя
20.05.2016	Склад	Склад	
03.06.2016	Юридический отдел	Начальник отдела	org_npo1

Рисунок В.5 – История перемещения устройства

Видеокарта ATI RADEON HD5400. Инв. № 0030			
Дата	Обслуживание	Описание	ФИО сотрудника
08.06.2016	Проверка работоспособности	Оборудование исправно	Димченко К. Г.

Рисунок В.6 – История обслуживания устройства

08.06.2016			
История обслуживания РМ Начальник IT-отдела (org_it1)			
Дата	Обслуживание	Описание	ФИО сотрудника
03.06.2016	Плановая проверка	Проверена работоспособность в	Калинин В.К.
12.06.2016	Определение уровня вредного о	Проверка показала, что уровень	Димченко К. Г.
21.06.2016	Замена монитора вследствие его	Заменен на монитор с инв. № 001	Калинин В.К.

Рисунок В.7 – История обслуживания рабочего места

				Форма по ОКУД по ОКПО		
(наименование организации)						
Сдатчик						
(наименование структурного подразделения)						
Получатель						
(наименование структурного подразделения)						
НАКЛАДНАЯ на внутреннее перемещение объектов основных средств				Номер документа	Дата составления 08.06.16	
Номер по порядку	Объект основных средств			Количество, шт.	Стоимость, руб.	
	наименование	дата приобретения (год выпуска, постройки)	инвентарный номер		единицы	всего
1	2	3	4	5	6	7
1	Видеокарта ATI RADEON HD5400	20.05.2016	0030	1	2000,00	2000,00

Рисунок В.8 – Накладная на внутреннее перемещение