



ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА

**Федеральное государственное бюджетное образовательное учреждение высшего образования
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)
Воронежский филиал**

Кафедра математики, информационных систем и технологий

Направление подготовки 09.03.02 Информационные системы и технологии
(код, наименование направления подготовки/специальности)

Форма обучения заочная

«К ЗАЩИТЕ ДОПУЩЕН(А)»
Заведующий кафедрой

(подпись)

Черняева С. Н.

(ФИО)

2024

Выпускная квалификационная работа

Обучающегося Селина Никиты Сергеевича
(фамилия, имя, отчество)

Вид работы выпускная квалификационная работа бакалавра
(выпускная квалификационная работа бакалавра, специалиста, магистра)

Пояснительная записка

Тема Разработка приложения для учета обслуживания материально-технической базы
предприятия (на примере АО «Воентелеком 127 военный завод»)
(полное название темы квалификационной работы, в соответствии с приказом об утверждении тематики ВКР)

Руководитель работы к. т. н., доцент Матыцина И. А. 21.06.24
(должность, подпись, фамилия, инициалы, дата)

Консультант _____
(при наличии) (должность, подпись, фамилия, инициалы, дата)

Консультант _____
(должность, подпись, фамилия, инициалы, дата)

Обучающийся Селин Н. С. 21.06.24
(подпись, фамилия, инициалы, дата)

Воронеж
2024

ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА

**Федеральное государственное бюджетное образовательное учреждение высшего образования
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)
Воронежский филиал**

Кафедра математики, информационных систем и технологий

Направление подготовки 09.03.02 Информационные системы и технологии

(код, наименование направления подготовки/специальности)

Форма обучения заочная

УТВЕРЖДАЮ
Заведующий кафедрой

(подпись)

Черняева С. Н.

(ФИО)

2024

**Задание
на выпускную квалификационную работу**

Вид работы ВКР бакалавра

(ВКР бакалавра, ВКР специалиста, ВКР магистра)

Обучающемуся Селину Никиту Сергеевичу

(фамилия, имя, отчество)

Тема Разработка приложения для учета обслуживания материально-технической базы
предприятия (на примере АО «Воентелеком 127 военный завод»)

Утверждена приказом ректора университета от _____ 20____, № _____

Срок сдачи законченной работы _____ 20____

Исходные данные (или цель ВКР):

Разработать приложения для учета обслуживания материально-технической базы
предприятия

Перечень подлежащих исследованию, разработке, проектированию вопросов (краткое содержание ВКР):

(актуальность темы, цели и задачи ВКР; аналитический обзор литературных источников; постановка задачи исследования, разработки, проектирования; содержание процедуры исследования, разработки, проектирования; обсуждение результатов; дополнительные вопросы, подлежащие разработке; заключение – выводы по работе в целом, оценка степени решения поставленных задач, практические рекомендации; и др.)

– Введение. Актуальность выбранной темы, цель и задачи ВКР

(наименование вопроса, раздела и его краткое содержание)

– Исследовательский раздел. Общая характеристика организации, экономические аспекты необходимости автоматизации задачи, существующая технология выполнения исследуемого процесса, постановка задачи на разработку информационной системы, анализ существующих разработок, описание видов обеспечения исследуемого процесса

(наименование вопроса, раздела и его краткое содержание)

– Проектный раздел. Обоснование выбора средств разработки, описание реализации базы данных, логическая модель, характеристика справочной и входной информации, характеристика базы данных, характеристика выходной информации для проектируемой системы, физическое моделирование информационной системы, описание программных модулей

(наименование вопроса, раздела и его краткое содержание)

– Заключение. Выводы по работе в целом. Оценка степени решения поставленных задач

(наименование вопроса, раздела и его краткое содержание)

Практические рекомендации

Перечень графического материала (или презентационного материала):

1. Титульный лист
2. Цель и задачи ВКР
3. Общая характеристика организации
4. Структура компании
5. Виды работ отдела
6. Диаграмма Учет заявок на техническое обслуживание «Как есть»
7. Диаграмма Учет заявок на техническое обслуживание «Как должно быть»
8. Постановка задачи
9. Сравнение существующих разработок
10. Выбор средств разработки
11. Реализации базы данных
12. Основные функциональные возможности системы
13. Реализация
14. Результаты ВКР

Консультанты по разделам ВКР (при наличии):

1. _____
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)
2. _____
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)
3. _____
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)

Дата выдачи задания: _____ 20____

Задание согласовано и принято к исполнению: _____ 20____

Руководитель ВКР: к. т. н., доцент Матьцина И. А. _____
(должность, ученая степень, ученое звание, ФИО) (подпись)

Обучающийся: _____ Селин Н. С. _____
(учебная группа, ФИО) (подпис)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ	8
1.1 Общая характеристика организации	8
1.2 Экономические аспекты необходимости автоматизации задачи	12
1.3 Существующая технология выполнения исследуемого процесса	13
1.3.1 Постановка задачи на разработку информационной системы	22
1.4 Анализ существующих разработок	26
1.5 Описание видов обеспечения исследуемого процесса	29
2 ПРОЕКТНЫЙ РАЗДЕЛ	34
2.1 Обоснование выбора средств разработки	34
2.1.1 Описание реализации базы данных	39
2.1.1.1 Логическая модель	39
2.1.1.2 Характеристика справочной и входной информации	43
2.1.1.3 Характеристика базы данных	44
2.1.1.4 Характеристика выходной информации для проектируемой системы	50
2.2 Физическое моделирование информационной системы	51
2.3 Описание программных модулей	52
ЗАКЛЮЧЕНИЕ	64
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	66
ПРИЛОЖЕНИЕ А	66
ПРИЛОЖЕНИЕ Б	67

ВВЕДЕНИЕ

Актуальность темы исследования обуславливается тем, что в современных условиях одной из важных задач на предприятиях является управление основными фондами. Необходимость автоматизации процесса управления активами вызвана потребностями ведения учета и обслуживания материально-технической базы, в которую входит большое количество оборудования и аппаратуры, а также контроля лицензий программных продуктов.

У многих людей, далеких от информационных технологий, образ системного администратора ассоциируется со специалистом по устранению сбоев компьютерной и оргтехники, реже – с человеком, ответственным за закупку и списание той самой техники. Однако подобные действия выполняются в соответствии с предписаниями бухгалтерии, контролирующей материальные активы компании. Именно на этом этапе возникает потребность в инвентаризации компьютеров и сопутствующих устройств.

На сегодняшний день для эффективной организации деятельности подразделений по учету вычислительной техники необходимо владение полной, точной и актуальной информацией о состоянии оборудования на предприятии. Однако большое количество и разнообразие технических объектов на производстве усложняет задачу по сбору и обработке данных, а также планированию технического обслуживания и ремонта. Применение современных информационных технологий позволит снизить трудоемкость выполнения этих процессов.

Объектом исследования настоящей выпускной квалификационной работы (ВКР) выступает организация АО «Воентелеком» многопрофильный телекоммуникационный холдинг.

Предметом исследования является процесс учет обслуживания материально-технической базы АО «Воентелеком».

Цель выпускной квалификационной работы состоит в разработке приложения для учета обслуживания материально-технической базы предприятия.

Для достижения цели, поставленной в работе, необходимо решить следующие задачи:

- выполнить анализ предметной области;
- провести анализ организации с целью выявления процессов учета, требующих автоматизации;
- провести анализ существующих разработок;
- провести обоснование выбора средств реализации проекта;
- описать функционал структуры и алгоритмов работы;
- спроектировать структуру базы данных;
- разработать приложение для учета обслуживания материально-технической базы предприятия;
- выполнить тестирование.

Структура выпускной работы представлена введением, двумя главами, заключением, списком использованной литературы и приложениями.

Для изучения теоретических основ объекта исследования, в первой главе проведено описание деятельности организации АО «Воентелеком», изучена структура, описаны технологии обработки информации с целью выявления недостатков существующего учета, был определен перечень задач, которые должна решать проектируемая система, и описаны требования.

В первой главе также проведен обзор и анализ наиболее популярных приложений от известных разработчиков для учета обслуживания операций и описана предлагаемая технология выполнения исследуемого процесса.

Во второй главе проведен обзор и анализ средств разработки клиентской части и систем управления базами данных с целью выбора

решения, способного удовлетворить требованиям, предъявленным к разрабатываемой системе и описано информационное обеспечение задачи.

Во второй главе описана логическая и физическая модель данных для исследуемой предметной области: приводится состав и структура базы данных, обозначены ключевые поля и связи между таблицами, представлено графическое отображение схемы базы данных и т.п.

1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

1.1 Общая характеристика организации

АО «Воентелеком» является многопрофильный телекоммуникационный холдинг. Компания входит в ТОП-20 крупнейших IT-компаний России.

Организация работает на российском рынке с 2002 года [14].

Основные направления деятельности:

1. Исследования современных телекоммуникационных технологий в интересах Минобороны РФ с целью изучения возможности применения в Вооруженных силах РФ;

2. Проведение научно-исследовательских и опытно-конструкторских работ в области создания новых образцов телекоммуникационного оборудования, автоматизированных и информационных систем и приложений;

3. Проектирование, строительство и реконструкция объектов, сетей и техники связи предприятий и органов военного управления Минобороны РФ;

4. Обслуживание, ремонт и утилизация средств связи, автоматизированного управления в интересах Вооруженных сил РФ, воинских формирований и органов государственной власти;

5. Расчет электромагнитной совместимости радиоэлектронных средств, обеспечение деятельности радиочастотных органов Минобороны РФ в области регулирования использования радиочастотного спектра;

6. Осуществление операторской деятельности по предоставлению услуг связи всем категориям пользователей, в том числе в закрытых военных городках;

7. Услуги по технической защите информации;

В состав АО «Воентелеком» входят:

- 17 Центральный проектный институт связи;
- 5 ремонтных заводов;
- 10 региональных отделений;
- 6 кабельных участков;
- 3 дочерних общества.

Региональные отделения АО «Воентелеком»:

- Калининградский филиал (772 ремонтный завод средств связи);
- Санкт-Петербургский филиал (521 центральный ремонтный завод средств связи);
- Рыбинский филиал (190 центральный ремонтный завод средств связи);
- Ивановский филиал (733 центральный ремонтный завод средств связи);
- Приволжско-Уральский филиал;
- Самарский филиал;
- Северо-Кавказский филиал;
- Новосибирский филиал;
- Хабаровский филиал;
- Приморский филиал (741 ремонтный завод средств связи).

Воронежский филиал АО «Воентелеком» располагается по адресу г. Воронеж, улица Краснознаменная, дом 165 А.

Основные виды деятельности Воронежского филиала:

- сервисное обслуживание и ремонт средств связи общевойскового назначения;
- сервисное обслуживание и ремонт техники радиоэлектронного подавления;
- сервисное обслуживание и ремонт шифровальной аппаратуры связи;
- комплексное оснащение объектов МО РФ (проект Цифровизация);
- ремонтные работы и техническое обслуживание ИТ-оборудования.

Дополнительные услуги:

- услуги по механические обработки изделий, токарные, фрезерные, сварочные, лакокрасочные и другие работы;
- техническое обслуживание и ремонт АБШ (ГАЗ, УРАЛ, КАМАЗ, ЗИЛ, МТЛБ, БТР-80) [14].

В таблице 1.1 представлены реквизиты организации.

Таблица 1.1 – Реквизиты АО «Воентелеком»

Полное наименование организации	Акционерное общество «Воентелеком»
Сокращенное наименование организации	АО «Воентелеком»
ОГРН	1097746350151
Юридический адрес	107014 г. Москва ул. Большая Оленья д. 15А стр. 1
Телефон и факс по юридическому адресу	Факс (+7 495) 609-51-52
Фактический адрес	107014 г. Москва ул. Большая Оленья д. 15А стр. 1
Телефон и факс по фактическому адресу	+7 (495) 609-50-05
Полное наименование учреждения банка	Банк ВТБ (ПАО)
Расчетный счет	40702810700260000236
Корреспондентский счет	30101810700000000187
БИК	044525187
ИНН	7718766718
КПП	366443001
ОКАТО	45263591000
Код организации ОКПО	07627860
Генеральный директор	Артамонов Александр Михайлович

Под организационной структурой управления понимается упорядоченная совокупность устойчиво взаимосвязанных элементов,

обеспечивающих функционирование и развитие организации как единого целого.

Для определения списка автоматизируемых задач, необходимо подробнее ознакомиться с организационной структурой организации и выявить перечень подразделений, задействованных в этих операциях. На рисунке 1.1 представлена организационная структура АО «Воентелеком».



Рисунок 1.1 – Структурная схема АО «Воентелеком»

Руководство организацией осуществляет генеральный директор, который: обеспечивает исполнение всех обязательств организации перед государственными бюджетами; решает финансово-экономические и производственно-хозяйственные вопросы организации; обеспечивает соблюдение всех, установленных законодательством, условий и требований по охране труда и здоровья сотрудников и т.п.

В отсутствие генерального директора его обязанности исполняет заместитель директора на которого выходит приказ.

АО «Воентелеком» основывается на процессе управления по соответствующим функциям, направленным на решение поставленных задач и достижение намеченных целей. С этих позиций структура управления

представляется в виде системы оптимального распределения функциональных обязанностей, прав и ответственности, порядка и форм взаимодействия между входящими в ее состав органами управления и работающими в них людьми.

1.2 Экономические аспекты необходимости автоматизации задачи

Объектом рассмотрения задачи является процесс учета обслуживания материально-технической базы, аппаратного и программного обеспечения.

В АО «Воентелеком» насчитывается 33 объекта автоматизации, над которыми возлежит контроль инженера по учету оборудования и программного обеспечения.

«Управление технического перевооружения и реконструкции, обслуживания и ремонта объектов электросетевого хозяйства» (Отдел техподдержки) выполняет следующие виды работ:

- осуществляет техническую поддержку сотрудников АО «Воентелеком»;
- следит за состоянием вычислительной техники на предприятии;
- производит обновление антивирусных и иных программных средств, используемых работниками компании в процессе решения поставленных задач;
- ведение учета установки оборудования и программного обеспечения на объектах автоматизации.

В связи с ростом интереса предприятий и организаций к продуктам компании и увеличением количества заказов, руководство АО «Воентелеком» приняло решение расширить штат сотрудников компании и привлечь к работе молодых сотрудников, способных привнести в компанию новые идеи и предложения.

Еще на стадии испытательного срока стало понятно, что новые сотрудники компании сталкиваются с множеством проблем в процессе освоения технических и программных средств, в процессе работы.

Решением подобных проблем занимается техподдержка, которая отвечает на возникшие вопросы и производит обучение «новичков».

Учитывая возросшую нагрузку на отдел, а также учитывая текущие задачи и сложности, с которыми ежедневно сталкиваются работники данного отдела, было принято решение – оптимизировать работу отдела путем внедрения информационной системы.

1.3 Существующая технология выполнения исследуемого процесса

Для описания существующей технологии учета в организации материально-технического обслуживания, построим структурно-функциональную диаграмму изучаемого процесса по методологии SADT (Structured Analysis and Design Technique) – методология структурного анализа и проектирования. Среди множества CASE средств, пригодных для реализации таких диаграмм, предпочтение было отдано продукту компании «Computer Associates» – «BPwin». Проектирование функциональных моделей будет производиться на основе стандарта IDEF0.

Контекстная диаграмма процесса «Учет заявок на техническое обслуживание» в АО «Воентелеком» представлена на рисунке 1.2.

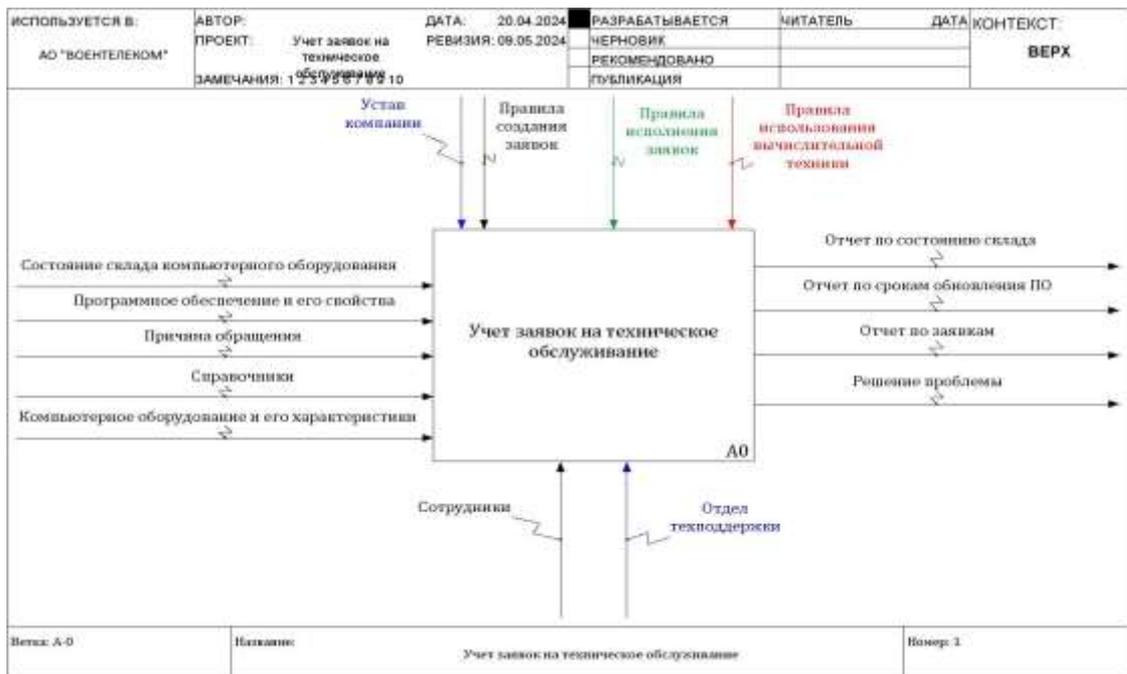


Рисунок 1.2 – Контекстная диаграмма процесса «Учет заявок на техническое обслуживание» в АО «Воентелеком». Модель "как есть"

На диаграмме дается общее представление изучаемого процесса.

Декомпозиция данного процесса представлена на рисунке 1.3.

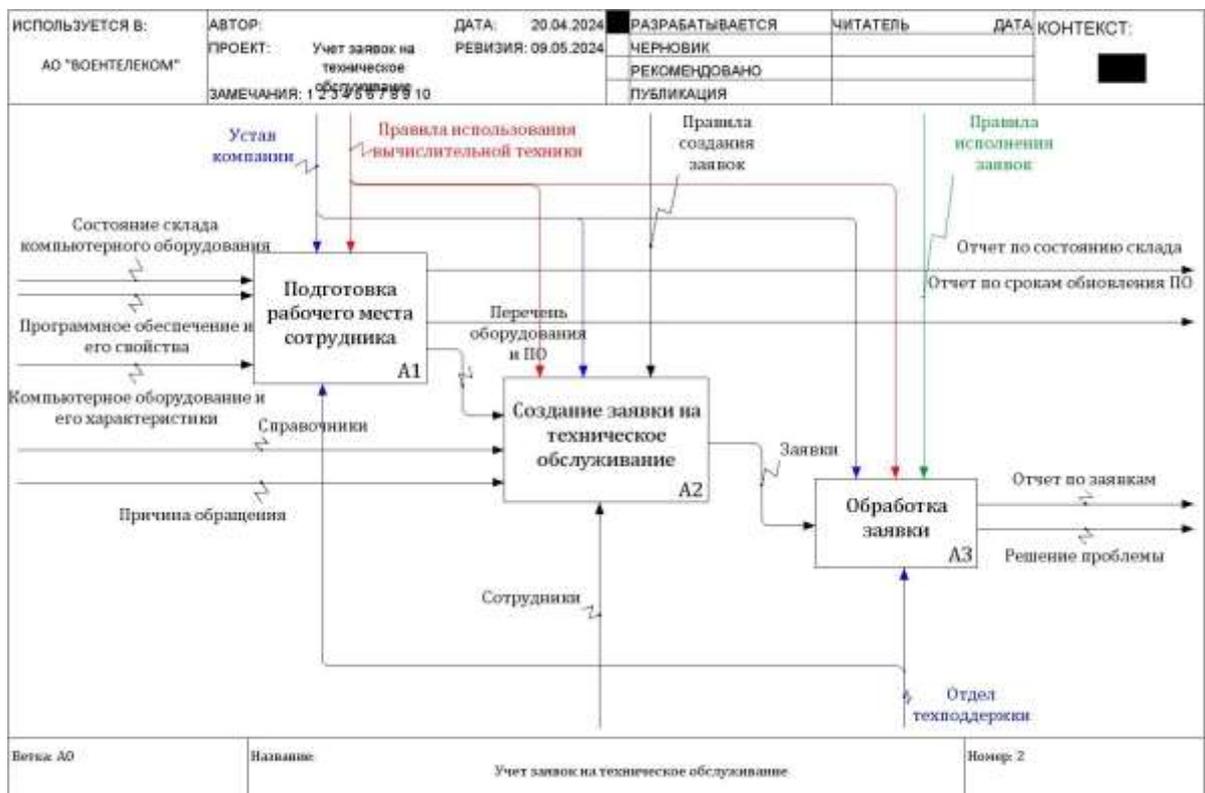


Рисунок 1.3 – Декомпозиция процесса «Учет заявок на техническое обслуживание» в АО «Воентелеком». Модель "как есть"

Рисунок 1.3 показывает, что операции по учету заявок на техническое обслуживание можно разбить на три блока, которые взаимосвязаны друг с другом:

- подготовка рабочего места сотрудника;
- создание заявки на техническое обслуживание;
- обработка заявки.

При выполнении изучаемого процесса формируются три отчета: «Отчет по состоянию склада», «Отчет по срокам обновления ПО», а также «Отчет по заявкам».

Рассмотрим каждый блок в отдельности. Для этого представим декомпозицию процесса «Подготовка рабочего места сотрудника», указанную на рисунке 1.4.

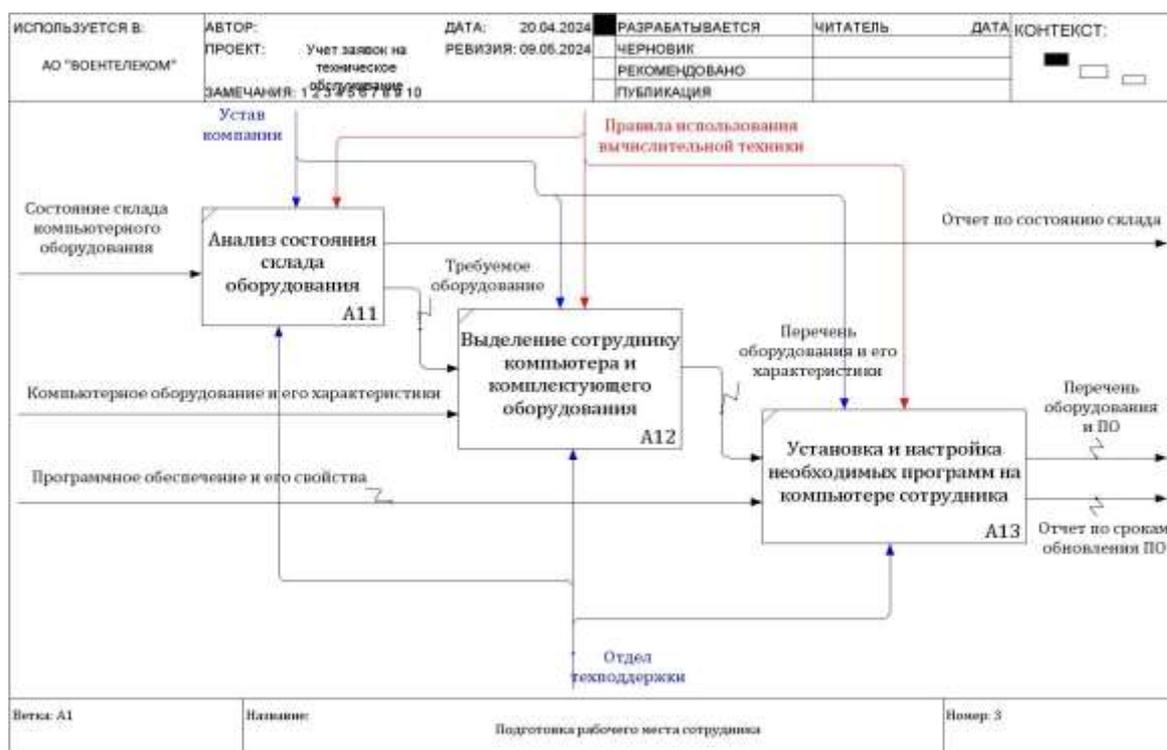


Рисунок 1.4 – Декомпозиция процесса «Подготовка рабочего места сотрудника». Модель "как есть"

Все операции в рамках процесса «Подготовка рабочего места сотрудника» осуществляются сотрудниками Отдела техподдержки, которые:

- проводят оценку состояния склада для поиска необходимого оборудования;

– на основе проведенного анализа, выделяют сотруднику компьютер и комплектующее оборудование, соответствующее должностным обязанностям сотрудника;

– проводят подготовку рабочего места сотрудника путем установки программного обеспечения (операционная система, антивирусная программа, пакет офисных программ и иное специальное ПО).

Перейдем к блоку «Создание заявки на техническое обслуживание», декомпозиция которого представлена на рисунке 1.5.

При возникновении вопроса, решение которого входит в компетенцию работников Отдела техподдержки, сотрудникам компании необходимо подготовить текст сообщения, содержащего суть возникшей проблемы и передать его в техподдержку. Обмен сообщениями при этом происходит в корпоративном чате компании.

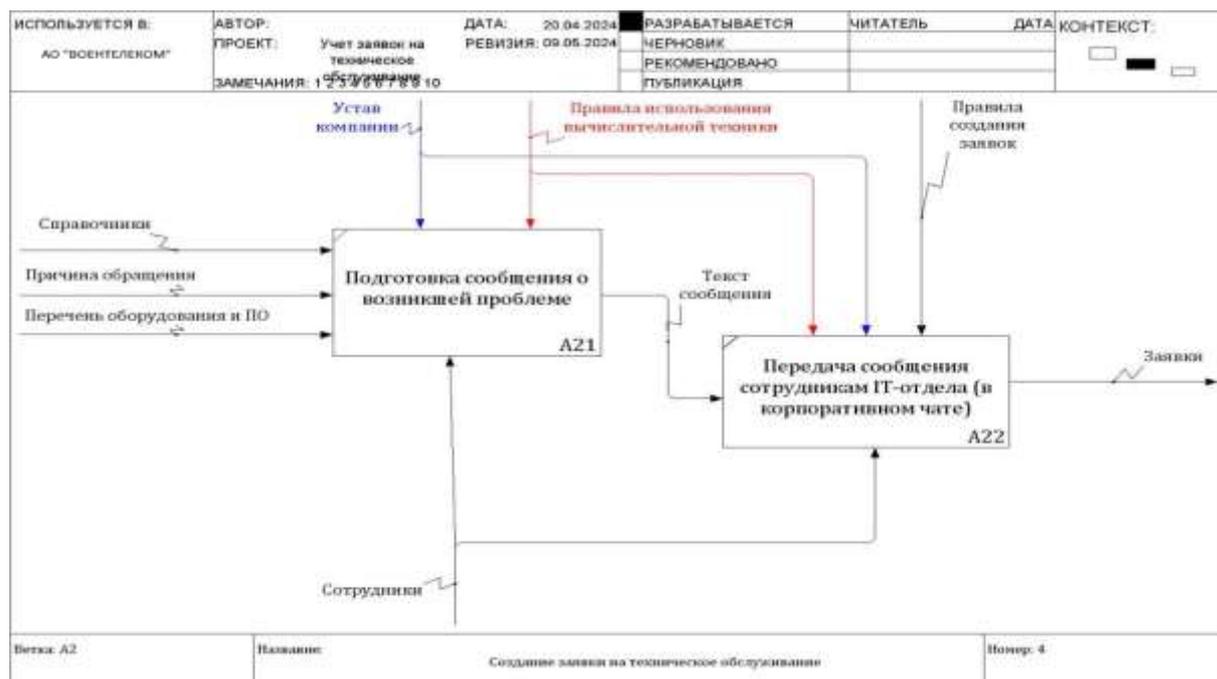


Рисунок 1.5 – Декомпозиция процесса «Создание заявки на техническое обслуживание». Модель "как есть"

На рисунке 1.6 представлена декомпозиция процесса «Обработка заявки».

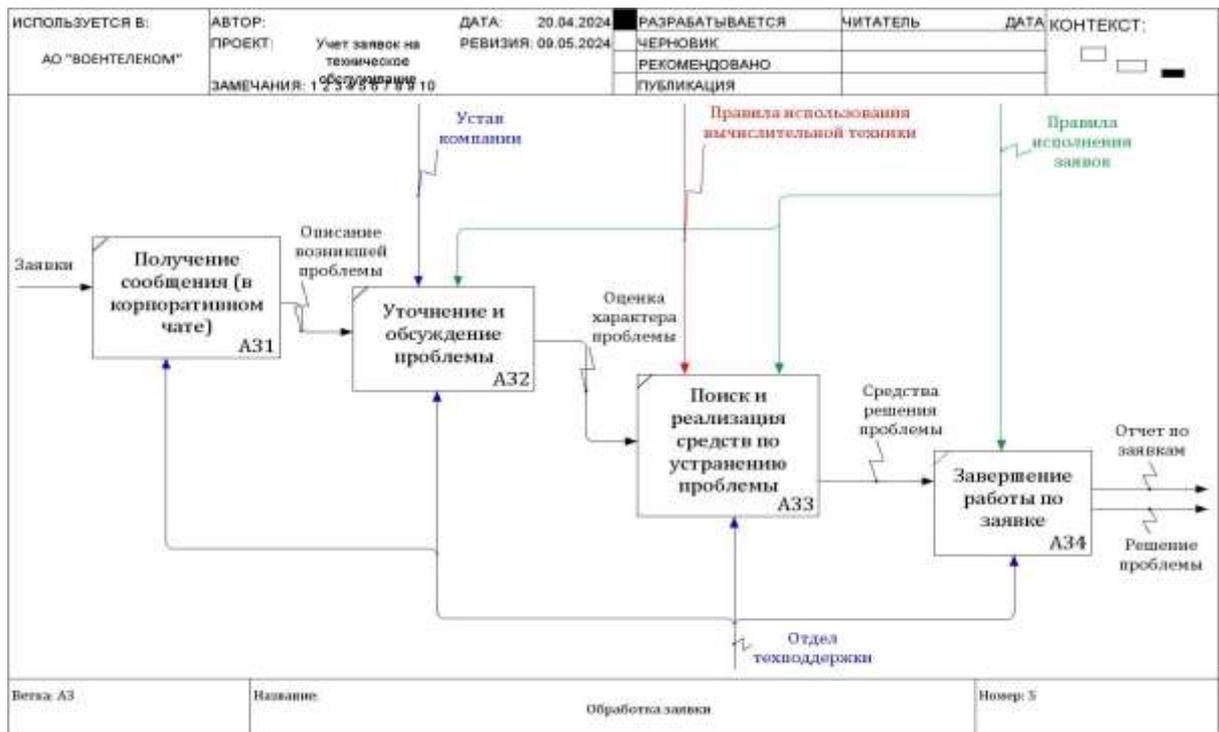


Рисунок 1.6 – Декомпозиция процесса «Обработка заявки». Модель "как есть"

Как показывает рисунок 1.6, при поступлении заявки на техническое обслуживание (в виде сообщения), сотрудник Отдела техподдержки, осуществляет следующие шаги:

- уточняет суть возникшего вопроса;
- производит поиск путей решения проблемы;
- сообщает о способе решения проблемы сотруднику, отправившему запрос.

Если сотрудника устраивает вариант решения проблемы, предложенный техподдержкой, то заявку можно считать закрытой.

Согласно построенной модели, существующая технология учета заявок на техническое обслуживание имеет множество недостатков:

- сотрудники Отдела техподдержки не имеют доступа к актуальной информации о состоянии компьютера и программного обеспечения пользователя, отправившего заявку, вследствие чего процесс обсуждения заявки становится довольно длительным;

– большую часть рабочего времени сотрудники техподдержки проводят в беготне по структурным подразделениям компании, где работники не могут продолжить работу без вмешательства IT–специалиста;

– низкая скорость обработки заявок приводит к застою в работе всех подразделений компании, а это конечным счетом сказывается на результатах деятельности самой организации;

– при создании текста заявки, сотрудники неправильно указывают названия программ, что снижает процесс восприятия вопроса, представленного в заявке;

– при возникновении вопросов, пути решения которых ранее предлагала техподдержка, сотрудникам приходится подолгу искать их в корпоративном чате, что не всегда удается, т.к. заявки не имеют таких параметров как «тема заявки», «название ПО» и т.п. Поиск усложняется, если сообщения были удалены;

– в связи с тем, что сотрудники удаляют сообщения, представленные в корпоративном чате, то при повторном возникновении аналогичной проблемы, техподдержке приходится повторно решать ее;

– при обращении сотрудника с вопросом о возможности замены комплектующего оборудования (клавиатуры или мыши), Отдел техподдержки не может дать однозначного ответа, т.к. не имеет доступа к актуальной информации о состоянии склада оборудования;

– сложность контроля состояния программного обеспечения пользователей – зачастую компьютеры остаются без защиты, т.к. антивирусная программа не была вовремя обновлена или продлена;

– сложность оценки результатов работы службы техподдержки – руководство компании не имеет перед собой механизма контроля деятельности работников отдела техподдержки;

– отсутствие механизмов, способных контролировать сроки исполнения заявок;

- отсутствие единых стандартов сбора и регистрации информации, приводящих к дублированию или потере данных о запросах;
- недостоверные результаты, представленные в отчетах.

В связи с этим, компании необходима информационная система, способная устранить существующие проблемы учета, повысить качество и скорость обработки заявок на техническое обслуживание.

На рисунке 1.7 изображена схема потоков данных операции «Подготовка рабочего места сотрудника» согласно предлагаемой технологии, т.е. после внедрения ИС.

Согласно предлагаемой технологии, сотрудникам техподдержки не придется самостоятельно проводить анализ состояния склада, чтобы выделить работнику компьютер и комплектующее оборудование – достаточно будет вызвать соответствующую форму, а система автоматически отразит в ней актуальную информацию о состоянии склада вычислительной техники компании.

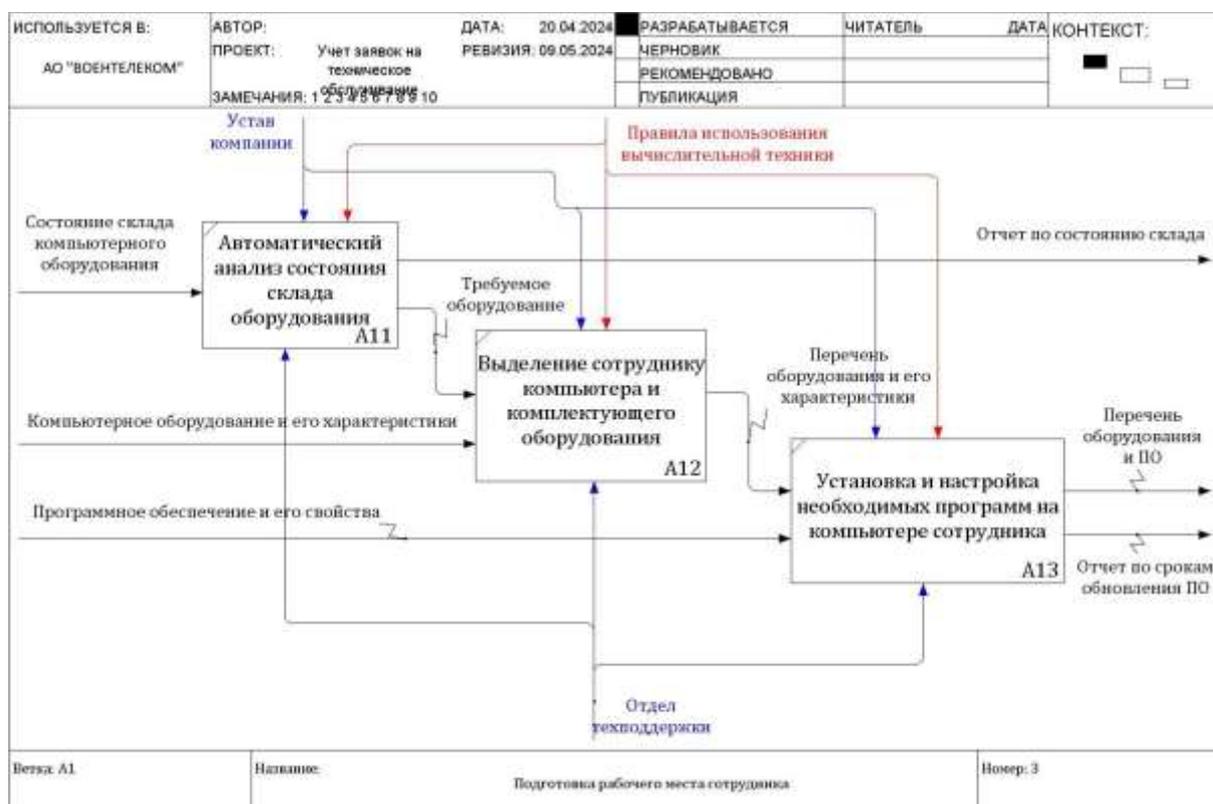


Рисунок 1.7 – Декомпозиция процесса «Подготовка рабочего места сотрудника». Модель "как должно быть"

Также упрощается и процесс формирования следующих отчетов: «Отчет по состоянию склада»; «Отчет по срокам обновления ПО».

Система будет формировать их автоматически на основе требований пользователя.

Процесс комплектации заказа заканчивается формированием документа «Упаковочный лист». Данный документ содержит список товаров, которые необходимо укомплектовать в рамках одного заказа. Отчет необходим фасовщикам, чтобы те имели перед собой перечень товаров (код, название товара, дата производства) и место их расположения (адрес ячейки хранения).

Диаграмма потоков данных показывает, что при движении информации от менеджера к сотрудникам склада, информация неоднократно дублируется, что в несколько раз замедляет процесс исполнения заказов.

На рисунке 1.8 изображена декомпозиция процесса «Создание заявки на техническое обслуживание».

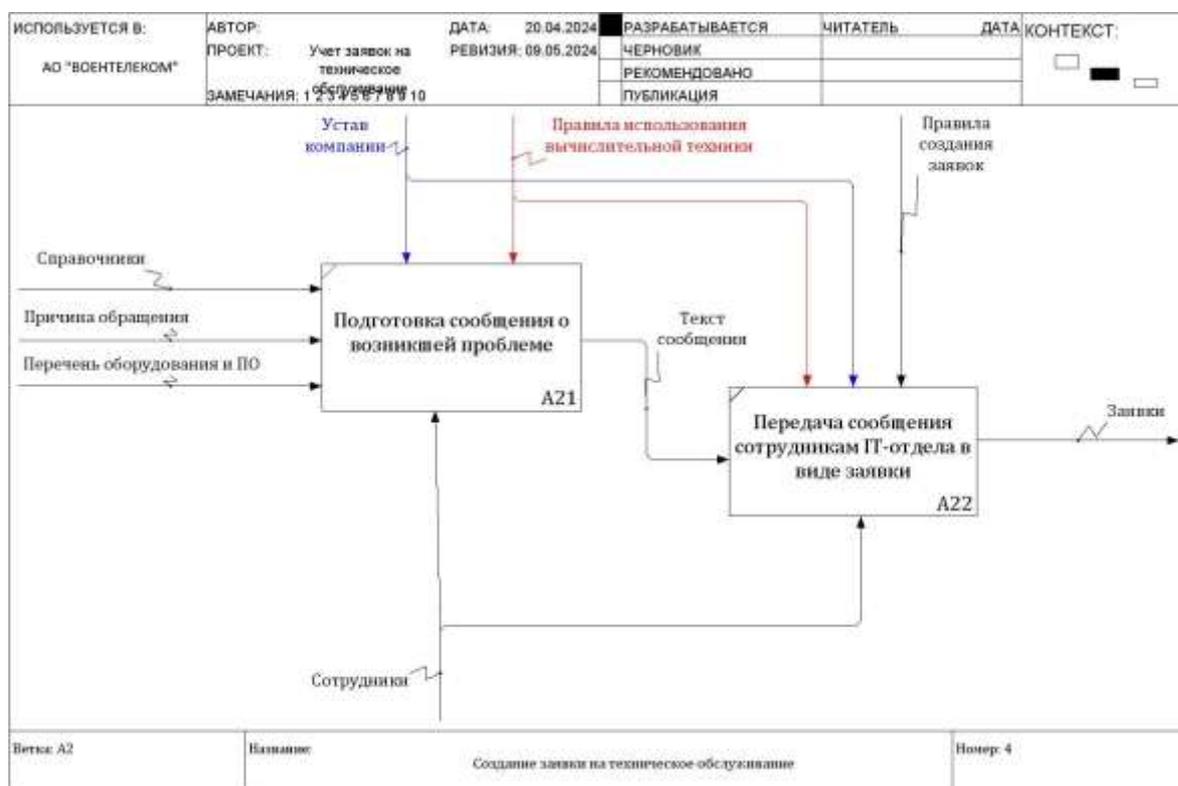


Рисунок 1.8 – Декомпозиция процесса «Создание заявки на техническое обслуживание». Модель "как должно быть"

Рисунок 1.8 показывает, согласно предлагаемой технологии, все заявки должны регистрироваться и храниться в проектируемой нами системе, а не в корпоративных чатах компании, как это было ранее.

Это позволит в несколько раз сократить время на формирование заявки за счет:

- автоматического определения системой значений некоторых полей;
- заполнения ряда полей путем выбора значений из списка.

На рисунке 1.9 изображена декомпозиция процесса «Обработка заявки».

Согласно новой технологии, сотрудники техподдержки получают заявки не посредством сообщений в корпоративном чате, а в виде полноценной заявки с номером, датой создания, с ФИО и отделом сотрудника, с темой заявки и названием ПО, относительно которого возникла проблема и т.п.

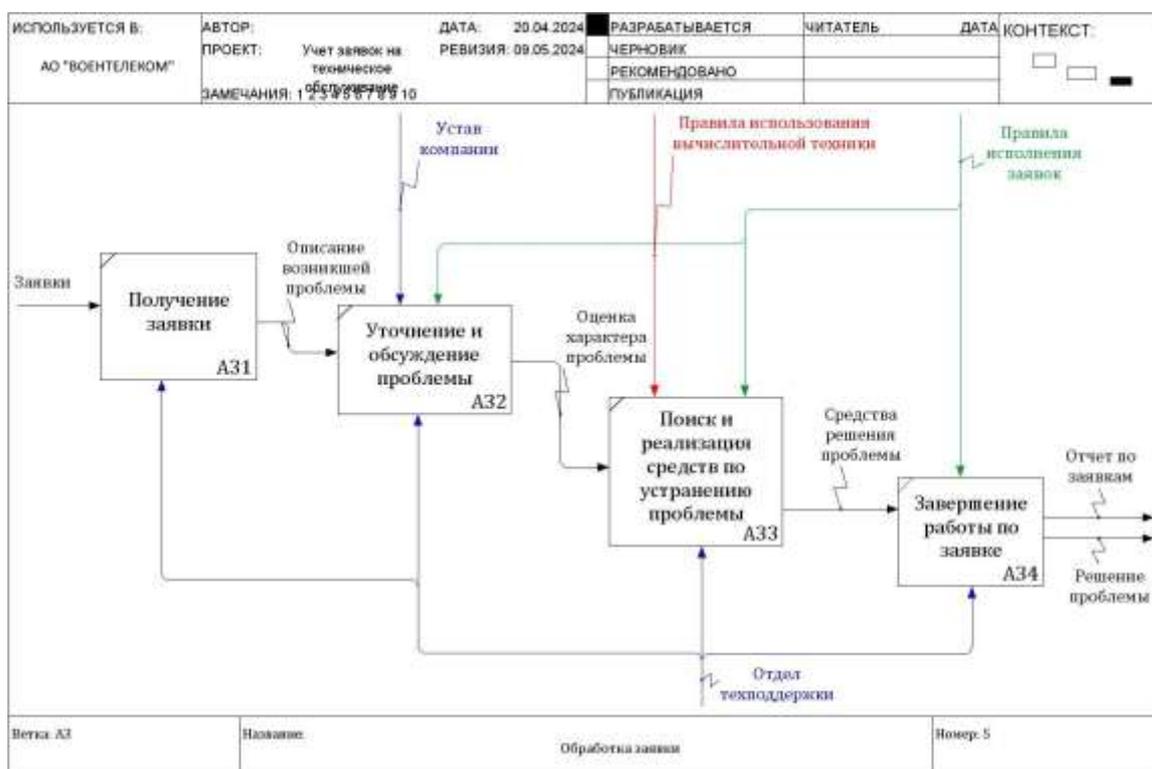


Рисунок 1.9 – Декомпозиция процесса «Обработка заявки». Модель "как должно быть"

На основе информации о зарегистрированных в системе заявках, программа будет автоматически формировать «Отчет по заявкам» за любой интервал времени.

Предложенная технология осуществления изучаемых бизнес-процессов позволит: устранить существующие недостатки учета; повысить скорость и качество обработки заявок; устранить ошибки связанные с человеческим фактором; сделать работу сотрудников компании более прозрачной и т.п.

1.3. Постановка задачи на разработку информационной системы

Основываясь на том, что в процессе учета заявок на техническое обслуживание участвуют все сотрудники компании, проектируемая информационная система должна состоять из двух модулей: «Системный администратор» и «Пользователь». Вход в систему должны иметь только зарегистрированные пользователи, прошедшие идентификацию по логину и паролю. Логин и пароль задаются системным администратором и, при необходимости, изменяются пользователем в «Личном кабинете». Такой принцип работы системы обеспечит защиту данных от недоброжелателей.

Основными функциональными возможностями модуля «Системный администратор» должны быть:

- прием, обработка и исполнение заявок;
- подготовка рабочего места сотрудника;
- учет сотрудников;
- добавление новых пользователей в систему;
- ведение справочников;
- учет используемого оборудования и их характеристик;
- учет программного обеспечения, установленного на каждом компьютере;

- контроль состояния склада технических средств;
- автоматическое формирование и вывод на печать отчетов;
- поиск и фильтрация записей.

Функция «Прием заявки» предполагает, что при создании в системе новой заявки через модуль «Пользователь» сотрудники Отдел техподдержки должен тут же увидеть ее в своем интерфейсе, так как в отделе работает не один сотрудник, то предусмотрена опция «Принять заявку в работу». После совершения данной операции, статус заявки меняется со значения «Новая» на «В работе», а в поле «Исполнитель» прописывается имя сотрудника, принявшего заявку в работу. Таким образом пользователь, отправивший заявку, сможет увидеть, что заявка уже находится в работе у определенного сотрудника.

«Обработка и исполнение заявки» означает, что после того как сотрудник принял заявку в работу, система должна позволить производить обмен сообщениями для уточнения сути возникшей проблемы и предложения мер по ее устранению. При этом любой обмен информацией, касающийся процесса исполнения заявки, должен производиться исключительно в системе, будь то обмен сообщениями или же графическими материалами.

Функция «Подготовка рабочего места сотрудника» должна позволить хранить в системе:

- перечень выделенного сотруднику оборудования (системный блок, монитор, манипулятор, клавиатура, наушники и т.п.) и их характеристики (цвет, производитель, модель и т.п.);
- перечень установленного на компьютер программного обеспечения и их свойств (дата установки, дата последнего обновления и дата следующего обновления, которая должна рассчитываться автоматически).

Функция учета сотрудников должна быть реализована через справочник «Сотрудники», где будет храниться следующая информация о

сотрудниках компании: ФИО, должность, дата рождения, адрес, номер телефона, e-mail, примечания. Также в данном справочнике должны задаваться логины и пароли для входа в систему.

Добавление новых пользователей в систему означает, что лишь у Администратора системы должен быть доступ к этой функции, чтобы избежать несанкционированного доступа к данным путем самостоятельной регистрации в системе.

Каждый из предусмотренных в системе справочников должен в несколько раз ускорить работу каждого сотрудника компании, за счет ввода заранее подготовленного в справочниках набора значений.

Функция «Учет используемого оборудования и их характеристик» должна позволить при поступлении новой заявки просмотреть информацию о характеристиках компьютера пользователя. Оперативный доступ к информации подобного характера в несколько раз ускоряет процесс исполнения заявок.

Функция «Учет программного обеспечения, установленного на каждом компьютере» необходима для просмотра сведений о программном обеспечении пользователя (название ПО, версия ПО), чтобы, не уточняя у пользователя дополнительных сведений, предложить меры по устранению возникшей проблемы.

Функция «Контроль состояния склада технических средств» должна позволить автоматически рассчитывать количество оборудования каждого вида, как оставшегося на складе, так и находящегося в пользовании у сотрудников.

Возможность ведения поиска записей позволит сократить время пользователя на поиск необходимой информации, а функция фильтрации – избавит пользователя от избыточной информации на экране. Функция фильтрации особенно полезна при формировании отчетов.

Автоматическое формирование и вывод на печать отчетов предполагает, что система, на основе заранее заполненных справочников, и

зарегистрированных пользователями системы данных должна автоматически формировать следующие виды отчетов:

- «Отчет по состоянию склада»;
- «Отчет по срокам обновления ПО»;
- «Отчет по заявкам».

Данные отчеты позволят проводить быстрый анализ результатов деятельности отдела техподдержки компании за текущий и прошедшие периоды времени.

Для сотрудников, работающих в системе через модуль «Пользователь» должны быть доступны следующие функции:

- регистрация, редактирование и удаление заявок;
- обмен графическими материалами;
- контроль статуса заявок;
- просмотр сведений о состоянии компьютера и об установленном на нем программном обеспечении;
- смена логина и пароля.

Регистрация заявок предполагает создание записи с указанием следующих данных:

- номер и дата заявки (определяются автоматически);
- ФИО и название отдела сотрудника (определяются автоматически);
- тема заявки;
- категория и название ПО, по которому возник вопрос;
- статус заявки (определяются автоматически) и т.п.

Все заявки, зарегистрированные в системе, должны иметь определенный статус, отражающий ход их исполнения. Целесообразно использование системы из 4 статусов: «Новая», «В работе», «Завершена», «Отмена».

Статус «Новая» автоматически присваивается новым заявкам. Заявки, которые находятся на исполнении у определенного сотрудника

техподдержки, должны иметь статус «В работе». Если работа по заявке завершена, то ей присваивается статус «Завершена». Статус «Отмена» имеют заявки, отмененные пользователем по какой-либо причине.

«Обмен графическими материалами» означает, что при создании заявки пользователь должен иметь возможность прикрепления файла, содержащего, например, скриншот возникшей ошибки. Данный файл будет доступен для просмотра сотруднику техподдержки.

Отслеживание статуса заявок означает, что пользователь должен в своем аккаунте видеть, на каком этапе находится исполнение заявки.

Использование системы статусов позволит также производить разграничение прав доступа пользователей к ним.

Функция «Просмотр сведений о состоянии компьютера и об установленном на нем программном обеспечении» должна предоставить пользователю доступ к характеристикам используемого им оборудования, перечню установленным за его рабочим местом программ и их свойствам.

Функция по смене логина и пароля предполагает изменение параметров входа в систему.

1.4 Анализ существующих разработок

Современный рынок программных продуктов предлагает предприятиям множество решений для автоматизации учета заявок на техническое обслуживание. Наибольшей популярностью среди них пользуются такие системы, как «ServiceNow», «HelpDesk», «1C: ServiceDesk» и «WorkFlowSoft». Данные системы находятся в свободном доступе и обладают минимальным набором функций.

Для оценки особенностей перечисленных программных продуктов и соответствия их функциональных возможностей требованиям, предъявленным компанией АО «Воентелеком», произведем сравнительный анализ вышеуказанных систем (таблица 1.2).

Таблица 1.2 – Сравнительная характеристика программных продуктов

Функциональные возможности	WorkFlowSoft	ServiseDesk	ServiceNow	HelpDesk
Прием, обработка и исполнение заявок	+	+	+	+
Подготовка рабочего места сотрудника	–	–	+	+
Учет сотрудников	+	+	+	+
Учет используемого оборудования и их характеристик	–	–	+/-	+/-
Контроль состояния склада технических средств	–	–	–	–
Формирование и вывод на печать отчетов	+	+	+	+
Поиск и фильтрация записей	+	+	+	+
Регистрация, редактирование и удаление заявок	+	+	+	+
Обмен сообщениями и графическими материалами	+	+/-	+	+
Контроль статуса заявок	+	–	+	+
Просмотр сведений о состоянии компьютера и об установленном на не ПО	–	–	–	+
Смена логина и пароля	+	+	+	+
Многопользовательский режим работы	+	+	+	+

Анализ приведенных в таблице данных свидетельствует, что все перечисленные системы обладают минимальным набором функций, необходимых для автоматизации учета заявок на техническое обслуживание, но у каждой из них есть свои недостатки.

Рассмотрим, какая из изучаемых программ больше остальных удовлетворяет требованию «цена-качество».

Программный продукт «WorkFlowSoft» от компании «WorkFlowSoft» не соответствует всем предъявленным требованиям к функциональности, но имеет невысокую стоимость. Так, система, рассчитанная на 5 пользователей, предоставляется бесплатно. Если требуется большее количество рабочих мест, то стоимость решения составит 700 рублей на пользователя в месяц [18].

В продукте компании «1С» – «ServiceDesk» отсутствуют базовый набор функций, необходимых для полноценного учета заявок. Интерфейс программы имеет сложную и непонятную структуру. Для внедрения такой системы придется нанимать программиста 1С и дорабатывать проект под нужды компании. С учетом того, что базовая версия программы предоставляется бесплатно, стоимость решения будет формироваться из зарплаты программиста (где-то от 5600 до 17000 руб.) [15].

«ServiceNow» от компании «ITSM365» является серьезным решением с большим набором функций. Стоимость решения варьируется в зависимости от количества необходимых лицензий. Версия программы, рассчитанная на 10 лицензий, обойдется 20000 рублей в месяц. Демоверсия программы предоставляется бесплатно [17].

Программа «HelpDesk» от компании «IT GUILT» – еще один достойный программный продукт для учета заявок на техническое обслуживание, поступающих от персонала компании. Стоимость решения составит от 5 до 25 тысяч рублей в месяц в зависимости от количества пользователей и регистрируемых заявок [16].

Анализ показывает, что информационные системы «HelpDesk» и «ServiceNow» больше остальных удовлетворяют требованиям, предъявленным АО «Воентелеком». Каждая из этих систем принесет немалую пользу компании после ее внедрения, однако нельзя забывать, что готовые решения не учитывают специфику работы конкретной организации, в связи с чем их не всегда удастся подстроить под ее бизнес-процессы.

Обычно есть два выхода из таких ситуаций: первый – подогнать бизнес-процессы компании под требования системы; второй – доработать внедряемую систему под нужды компании. И в первом и во втором случае есть свои минусы. Так, например, изменения в бизнес-процессах компании очень часто сопровождаются негативным настроем сотрудников, нежелающих менять привычную схему работы. Что касается второго варианта, то тут две проблемы, вытекающие одна из другой. Первая – компания-

разработчик может отказаться от доработок проекта и предложит использовать систему в готовом виде. Вторая проблема – большие финансовые затраты, необходимые для доработки системы.

Помимо этого, есть риск, что даже после доработки проекта, готовое решение не будет приносить желаемый результат в связи с растущими требованиями появляются новые идеи, задачи, функции, которые компания захочет видеть в используемой системе. К тому же внедрение готовых решений не гарантирует обеспечение безопасности данных, т.к. они находятся на стороннем сервере.

Итоги анализа существующих на отечественном рынке программных продуктов для автоматизации учета заявок на техническое обслуживание доказывают, что наиболее рациональным решением для АО «Воентелеком» является разработка собственной информационной системы, специально ориентированной под организацию.

1.5 Описание видов обеспечения исследуемого процесса

Организационное обеспечение включает в себя методы и средства взаимодействия сотрудников организации с аппаратными средствами и между собой в процессе работы с проектируемой информационной системой.

Взаимодействие пользователя с компьютерами и периферийными устройствами будет осуществляться посредством дружественного, графического пользовательского интерфейса и элементов управления, расположенных на самих технических устройствах (кнопки включения и отключения и т.п.).

Средствами, обеспечивающими взаимодействие пользователя с техническими устройствами, могут служить экран монитора, операционная система, проектируемая информационная система, окна проектируемой информационной системы.

Взаимодействие пользователя с проектируемой информационной системой может осуществляться посредством элементов управления, расположенных в окнах программы (главное меню, контекстное меню, кнопки, вкладки) и «горячих клавиш».

В качестве метода взаимодействия сотрудников с аппаратными средствами может служить совокупность действий пользователя над информационной системой для решения поставленных задач. Такими методами могут быть операции по добавлению, редактированию, удалению, обновлению, фильтрации, печати либо поиска данных и т.п.

В состав организационного обеспечения ИС входят структурные подразделения организации (пользователи), осуществляющие поддержку ее работоспособности и документы, регламентирующие порядок эксплуатации и развития системы. Такими документами могут служить «Техническое задание на разработку проекта», «Инструкции пользователей» и иные документы, которые будут представлены в пункте «Правовое обеспечение» данной работы.

Для работы с системой и поддержания ее работоспособности, необходимы следующие лица:

- администратор БД – лицо, ответственное за сопровождение данных, определение уровней доступа к ним, добавление/удаление пользователей БД и т.п. Обычно администраторы БД участвуют в процессе разработки структуры БД;

- администратор ИС – лицо, обладающее правами по: редактированию справочников системы; просмотру и выводу на печать отчетов, формируемых системой; изменению параметров доступа пользователей к системе; добавлению/удалению пользователей ИС;

- конечные пользователи – лица, работающие с ИС и БД через пользовательский интерфейс для решения поставленных задач.

Программное обеспечение.

Проектируемая система будет состоять из двух модулей: «Системный администратор» и «Пользователи».

Минимальное количество пользователей системы должно равняться 3 сотрудникам (по одному для каждого модуля + Администратор БД). Максимальное количество сотрудников, использующих систему, не ограничено.

Администратор БД должен обладать знаниями:

- проектирования и сопровождения БД;
- обеспечения целостности и безопасности хранимых данных;
- определения уровней доступа к данным;
- добавления/удаления пользователей БД;
- резервного копирования и восстановления данных.

Остальные пользователи, работающие с системой должны обладать знаниями в исследуемой предметной области и навыками пользования компьютером на уровне продвинутого пользователя.

Надежность системы должна обеспечиваться путем:

- применения программно-аппаратных средств, соответствующих классу решаемых задач;
- соблюдения правил эксплуатации и технического обслуживания оборудования;
- предварительного обучения пользователей при работе с программой.

В случае возникновения сбоев в электроснабжении, восстановление системы должно осуществляться сразу после перезагрузки операционной системы и запуска исполняемого файла.

Все ошибки, ведущие к сбоям в программе и неточности в расчетах, выявленных на этапах тестирования и внедрения программного продукта, будут устраняться разработчиком по мере их обнаружения.

Проектируемая система должна соответствовать следующим требованиям к эргономике и технической эстетике:

- весь текст, отображаемый на экране, должен быть представлен на русском языке;

- функции по работе с базой данных (добавить, изменить и удалить) должны быть легкодоступными;

- ИС должна иметь дружелюбный, интуитивно-понятный интерфейс;

- меню программы не должно быть перегружено (не более 3 уровней);

- интерфейс программы должен быть выполнен в едином стиле с использованием приятной цветовой гаммы, не раздражающей зрение пользователя. Допускается такое сочетание цветов, как белый, серый, голубой или белый, серый, зеленый;

- стиль и размер шрифта должен быть читаемый (желательно использовать тип шрифта «Times New Roman» либо «Cambria» размером в 12-14 пт.);

- система должна предупреждать пользователя о выполнении им некорректных действий над программой путем выдачи на экран монитора сообщения с рекомендациями по устранению возникшей проблемы.

Требование к информационной безопасности должно быть обеспечено путем:

- предоставления доступа к системе пользователям, прошедшим идентификацию по логину и паролю (логин и пароль задаются Администратором и, при необходимости, изменяются пользователем в «Личном кабинете»);

- использованием лицензионного программного обеспечения;

- регулярной проверкой ПО на наличие компьютерных вирусов.

Техническое обеспечение – это комплекс технических средств, необходимых для функционирования информационной системы.

Для обеспечения работы проектируемой информационной системы необходим совместимый персональный компьютер, включающий в себя:

- процессор «Intel Core i5-4200U» с частотой 1.6 ГГц, не менее;

- оперативную память объемом, 4 Гб, не менее;
- жесткий диск объемом 500мб, не менее;
- операционную систему «Windows 7», не ранее;
- манипулятор «мышь»;
- стандартная клавиатура.

Требования к программному обеспечению: сервер баз данных «MySQL» и стандартный набор офисных программ «MicrosoftOffice».

2 ПРОЕКТНЫЙ РАЗДЕЛ

2.1 Обоснование выбора средств разработки

Процесс проектирования информационной системы начинается с выбора программных средств ее реализации: среды разработки и системы управления базами данных (СУБД).

В наше время многообразие языков программирования и средств разработки баз данных дает возможность любому разработчику, на основе личных предпочтений и функциональных возможностей, закладываемых в проектируемую информационную систему, производить выбор той среды разработки, которая окажется наиболее приемлемой для решения поставленных задач (начиная с разработки простых приложений и заканчивая серьезными программными продуктами для крупных корпораций).

Обоснование выбора СУБД.

Для выбора системы управления базами данных необходимо провести сравнительный анализ наиболее популярных СУБД («MS Access», «Visual FoxPro», «MySQL» и «Oracle») на предмет соответствия основным функциям, которые она должна выполнять: создание БД; манипулирование данными; физическое размещение и сортировка записей; защита от сбоев, поддержку целостности данных и их восстановление; безопасность данных [28].

СУБД «MS Access» превосходит своих оппонентов по простоте и удобству использования, но значительно уступает в средствах администрирования, быстродействии (особенно при больших объемах данных) и безопасности хранимых данных. «MS Access» рекомендуется использовать при проектировании несложных программ с ограниченным объемом записей [28].

«Visual FoxPro» – СУБД, работающая на собственном языке программирования. Недостатки системы такие же, как и у «MS Access».

Такую СУБД целесообразно использовать при создании сложных приложений с количеством записей не более миллиона [28].

«MySQL» – СУБД, полностью удовлетворяющая требованиям безопасности, надежности, компактного хранения и скорости обработки данных с мощными средствами администрирования БД. В отличие от предыдущих СУБД, «MySQL» позволяет хранить колоссальные объемы данных, проводить архивирование и восстановление БД. Несмотря на множество доступных функций, процесс управления СУБД не отличается высокой сложностью управления и использования. «MySQL» зарекомендовал себя как незаменимое средство для создания очень больших (централизованных или распределенных) баз данных для средних и крупных предприятий, где защита и сохранность данных – первоочередная задача [28].

В СУБД «Oracle» заложены такие же функции, что и в «MySQL», но, в отличие от последней, «Oracle» имеет высокую стоимость и сложные механизмы управления. Такую СУБД целесообразно применять в очень крупных предприятиях.

На рисунке 2.1 приводится экспертная оценка функционала рассматриваемых СУБД: «MS Access», «Visual FoxPro», «MySQL» и «Oracle» (по десятибалльной шкале [28]).

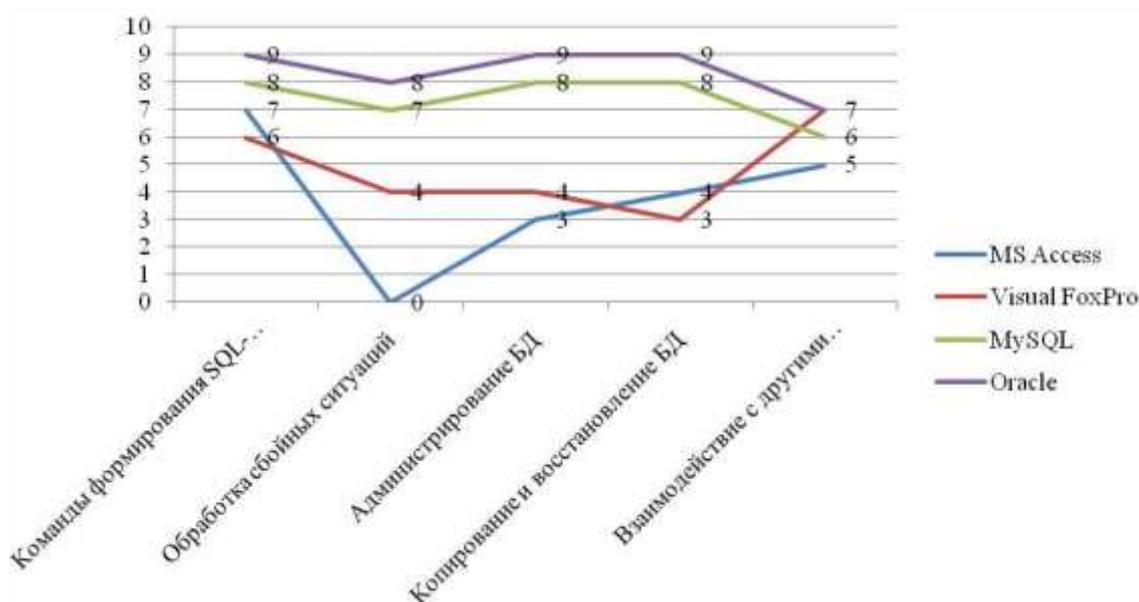


Рисунок 2.1 – Экспертная оценка основных СУБД

Итоги анализа СУБД показывают, что для разработки надежной и безопасной информационной системы целесообразно использовать «MySQL», т.к. в отличие от «Oracle», она является бесплатной и проста в освоении и использовании.

Обоснование выбора средств разработки клиентской части ИС.

В последние годы большой популярностью у программистов пользуются среды быстрой разработки приложений (RAD). К семейству таких систем программирования относятся: «Visual Basic», «Delphi», «C++», «C++ Builder», «C#», и другие [31].

В нашей работе выбор среды разработки будет производиться на основе анализа перечисленных выше языков программирования.

«Visual Basic» хорошо подходит для быстрой разработки простых и недорогих информационных систем. Достоинством «Visual Basic» можно назвать простую и удобную среду разработки, позволяющую работать с различными типами баз данных. Программы, написанные на «Visual Basic», отличаются невысокой скоростью выполнения и требуют подключения дополнительных DLL- файлов [31].

«Delphi» является мощной и быстрой средой разработки приложений на основе технологии объектно-ориентированного программирования (ООП).

Среди преимуществ «Delphi» можно выделить:

- отсутствие необходимости в установке дополнительных библиотек;
- встроенный инструмент визуального моделирования, реализующий графическую архитектуру, которая позволяет управлять целыми бизнес-объектами;
- мощный генератор отчетов;
- быстрый доступ к различным базам данных;
- огромное количество встраиваемых компонентов.

Язык «Delphi» по возможностям значительно превосходит язык «Visual Basic» и даже язык «C++» (например, «Delphi» более надежен, стабилен и легок в изучении) [31].

Язык «C++» позволяет эффективно использовать практически все возможности современных вычислительных средств и операционных систем. Это достаточно мощный язык, позволяющий создавать кроссплатформенные приложения любого назначения. «C++» считается относительно сложным языком программирования в связи с необходимостью запоминания методов работы с каждым объектом. Еще одним важным недостатком данного языка является синтаксис, провоцирующий ошибки и плохая поддержка модульности (при подключении большого количества модулей, серьёзно замедляется работа компилятора и заметно увеличивается размер конечного .exe файла) [31].

«C++ Builder» представляет собой некую интеграцию «Delphi» и «C++»: т.е. используется та же среда программирования (RAD), взятая из «C++» и библиотека визуальных компонентов (VCL), взятая из «Delphi». В этом и заключается основной недостаток этого языка – библиотека VCL, реализованная в «Delphi», существенно увеличивает размер исполняемых файлов. Преимущество «C++ Builder», по сравнению с предшественниками улучшенная версия объектной модели [31].

«C#» – язык, где сочетаются простота «Visual Basic» с мощностью «C++». Синтаксис языка аналогичен «C++», но более мощный, безопасный и простой. «C#» поддерживает работу с разными типами данных (а также позволяет создавать новые типы данных), статическую типизацию, перегрузку операторов и многое другое. Данный язык может применяться как при разработке приложений, так и для создания сайтов. В «C#», также как и у его оппонентов, есть свои недостатки, среди которых следует выделить моноплатформенность (разработка приложений исключительно для Windows) и низкое быстродействие [31].

На рисунке 2.2 приводится экспертная оценка возможностей рассматриваемых систем: «Visual Basic» (VB), «Delphi» (Del), «C++» , «C++ Builder» (CB) и C# (по десятибалльной шкале) [31].

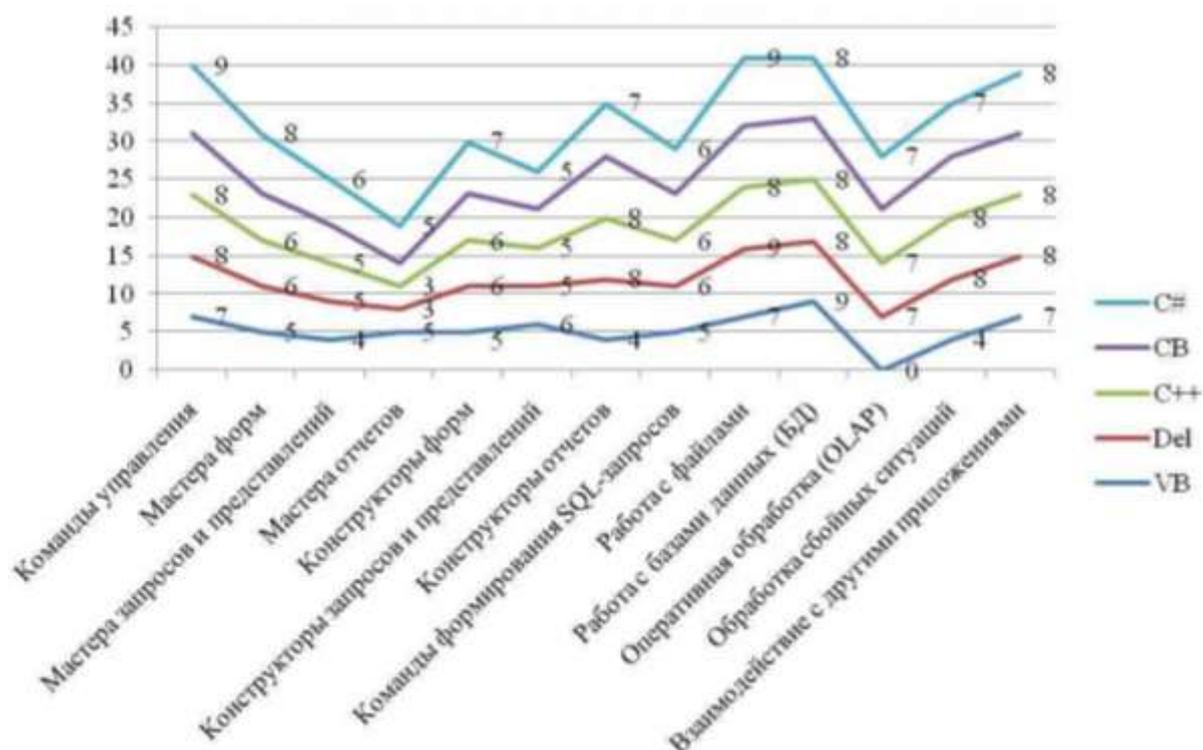


Рисунок 2.2 – Экспертная оценка основных средств разработки приложений

Таким образом, подводя итоги анализа языков программирования, становится очевидным, что каждый из них может быть успешно использован для разработки, проектируемой в данной работе, информационной системы. Поэтому, выбор среды программирования будет производиться на основе опыта, уровня знаний и предпочтений разработчика, т.к. они способны существенным образом отразиться и на качестве реализуемого программного продукта.

Для разработчика, наиболее понятным и удобным оказался язык программирования «Delphi». Наличие в интернете множества форумов по данной тематике, бесплатных обучающих ресурсов (книг, видео-уроков), обращение к которым поможет разобраться с возникшими в процессе программирования вопросами, только доказывают правильность принятого решения.

2.1 Описание реализации базы данных

2.1.1 Логическая модель

Логическая модель данных – графическое представление структуры базы данных. Она отображает перечень сущностей, их атрибутов, связей между ними и ограничений, налагаемые предметной областью [20].

Логическая модель данных составляется без привязки к определенной СУБД и является начальным прототипом будущей базы данных.

В настоящее время в области проектирования информационных систем с успехом применяется визуальное моделирование с помощью унифицированного языка моделирования UML.

В нашем случае, БД будет реализована в среде «MySQL» с использованием инструментария «dbForge Studio», представленного мощными средствами разработки и администрирования реляционных БД. Немаловажным достоинством «dbForge Studio» является возможность строить логические модели данных (схемы БД) на основе заранее созданных объектов.

Для создания логической модели данных, нам необходимо построить диаграмму вариантов использования и классов.

Диаграмма вариантов использования – это диаграмма, на которой изображаются отношения между актерами и вариантами использования, представлена на рисунке 2.3.

Диаграмма классов является частью логической модели системы и представляет статическую картину системы, представлена на рисунке 2.4.

Одним из подходов к построению модели базы данных является методология «IDEF IX», основанная на концепции «сущность-связь».

Сущность на диаграмме обозначается в виде прямоугольника, разделенного на две части: ключевая область, представленная ключевыми атрибутами; область данных, состоящую из неключевых атрибутов.

Ключевыми считаются атрибуты, имеющие первичный ключ. Все остальные атрибуты относятся к неключевым (в том числе и атрибуты, составляющие внешний ключ).

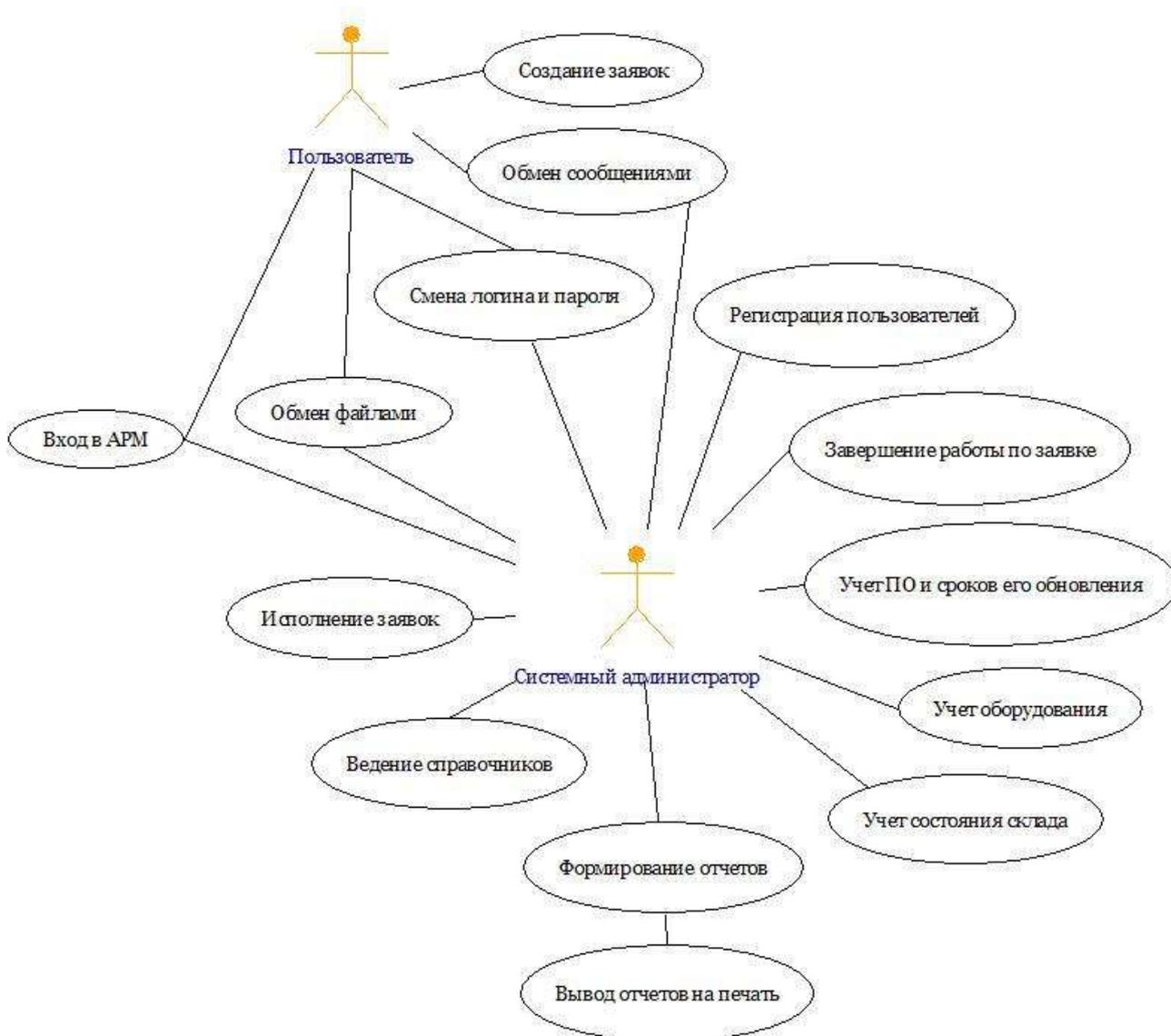


Рисунок 2.3 – Диаграмма вариантов использования

Атрибуты – перечень имен, расположенных внутри блока сущности. Связи или отношения между атрибутами изображаются в виде линии, соединяющей родительский атрибут с дочерним. Отношения показывают, по какому из существующих типов связи («один-ко-многим», «один-к-одному» или «многие-ко-многим») соотносятся атрибуты между собой. В нашем случае, используется связь «один-ко-многим», в связи с чем, один из концов линии связи будет представлен точкой [17].

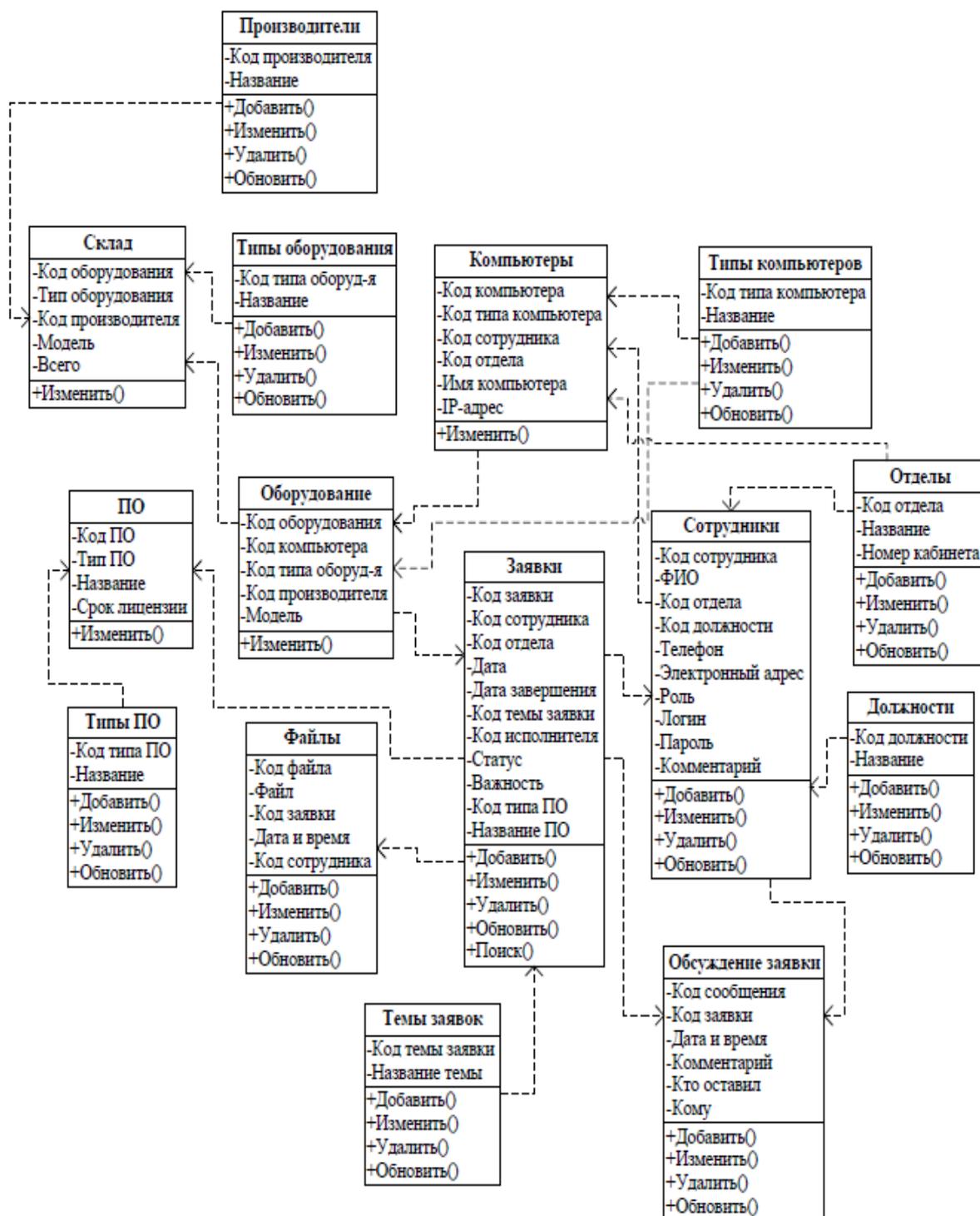


Рисунок 2.4 – Диаграмма классов

Логическая модель данных бизнес-процесса «Учет заявок на техническое обслуживание» на основе методологии «IDEF IX» представлена на рисунке 2.5

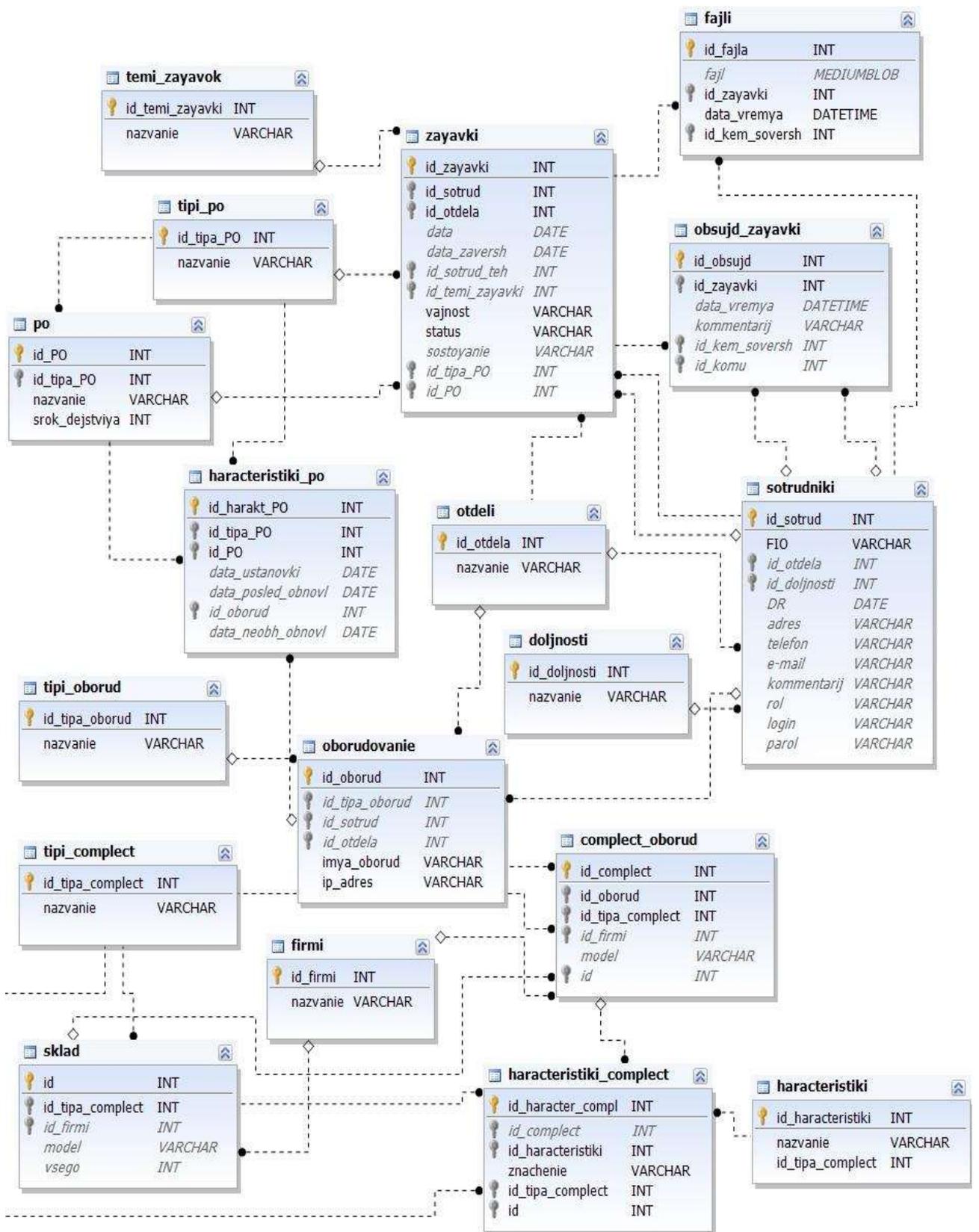


Рисунок 2.5 – Логическая модель данных

Модель базы данных, изображенная на рисунке 2.5, показывает, что проектируемая БД состоит из 18 таблиц, детальная информация о которых будет приведена при описании физической модели данных.

2.1.2 Характеристика справочной и входной информации

Входная информация – это данные, поступающие в систему извне. Для изучаемого бизнес-процесса такими источниками входных данных могут служить справочники системы: «Типы компьютеров», «Типы оборудования», «Характеристики типов оборудования», «Типы ПО», «ПО», «Производители», «Должности», «Отделы», «Сотрудники», «Темы заявок».

На основе информации из перечисленных справочников осуществляются все процессы по учету заявок на техническое обслуживание. Рассмотрим назначение каждого из перечисленных ранее справочников и таблиц.

Справочник «Типы компьютеров» хранит перечень типов компьютеров, имеющихся в компании. Например, персональный компьютер, ноутбук, моноблок и т.п.

Справочник «Типы оборудования» содержит названия технических устройств, необходимых для подготовки рабочего места сотрудника. Например, системный блок, монитор, клавиатура, манипулятор, наушники и т.п.

Справочник «Характеристики типов оборудования» необходим для хранения перечня возможных характеристик, определяющих каждый тип оборудования. Для клавиатуры такими характеристиками могут быть: цвет; подсветка; интерфейс связи с ПК (т.е. тип разъема); количество клавиш; тип клавиатуры; и т.п.

Справочник «Типы ПО» позволяет вести список типов программного обеспечения. Например, офисные программы, антивирусные программы, ПО специального назначения и т.п.

Справочник «ПО» представляет собой перечень программных продуктов с указанием их типа. Тип ПО, соответственно определяется из справочника «Типы ПО». Например, в данном справочнике может

содержаться запись: Название ПО: «Microsoft Word 2010»; Тип ПО: «Офисные программы».

Справочник «Производители» хранит названия известных производителей вычислительных средств и оборудования.

Справочник «Должности» необходим для ведения списка должностей сотрудников компании.

Справочник «Отделы» содержит перечень структурных подразделений организации.

Справочник «Сотрудники» служит для хранения информации о работниках компании: ФИО, отдел, должность, адрес, телефон, E-mail и данных для входа в систему (логин и пароль). Значения полей «Должность» и «Отдел» заполняются в соответствии со значениями из предыдущих двух справочников.

Справочник «Темы заявок» позволяет создавать и хранить перечень тем, по которым могут создаваться заявки.

2.1.3 Характеристика базы данных

Физическая модель данных информационной системы представляет собой логическую модель данных, выраженную средствами конкретной СУБД. В качестве средства разработки БД будет использована реляционная система управления базами данных «MySQL».

При создании физической модели данных отношения, описанные в логической модели, преобразуются в таблицы, атрибуты становятся столбцами, каждому атрибуту присваивается тип данных, а для ключевых столбцов создаются уникальные индексы [6].

Как было отмечено ранее, проектируемая БД состоит из 18 таблиц, (сущностей): «Оборудование», «Должности», «Файлы», «Производители», «Характеристики типов оборудования», «Характеристики оборудования», «Характеристики ПО», «Склад», «Сотрудники», «Темы заявок», «Типы

оборудования», «Типы компьютеров», «Типы ПО», «Заявки», «Обсуждение заявки», «ПО», «Фирмы» и «Отделы».

Атрибуты сущности «Оборудование» представлены в таблице 2.1.

Таблица 2.1 – Атрибуты сущности «Оборудование»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_complect	int(10) unsigned	NO	PRI	auto_increm.	Код оборудования
id_oborud	int(10) unsigned	NO	MUL	0	Код компьютера
id_tipa_complect	int(10) unsigned	NO	MUL		Код типа оборуд.
d_firmi	int(10) unsigned	YES	MUL		Код производителя
model	varchar(15)	YES	MUL		Модель
id	int(10) unsigned	YES	MUL		Код для связи

Атрибуты сущности «Должности» представлены в таблице 2.2.

Таблица 2.2 – Атрибуты сущности «Должности»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_doljnosti	int(10) unsigned	NO	PRI	auto_increm.	Код должности
Nazvanie	varchar(15)	NO	MUL		Название

Сущность «Файлы» представлен номер заявки, дата и время, а также кем выполнена работа. Атрибуты сущности «Файлы» приведены в таблице 2.3.

Таблица 2.3 – Атрибуты сущности «Файлы»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_fajla	int(10) unsigned	NO	PRI	auto_increm.	Код файла
fajl	mediumblob	YES			Файл
id_zayavki	int(10) unsigned	NO	MUL	0	Код заявки
data_vremya	datetime	NO		0000-00-00 00:00:00	Дата и время
id_kem_oversh	int(10) unsigned	NO	MUL	0	Код сотрудника

Сущность «Производители» представлен наименование производителя. Атрибуты сущности «Файлы» приведены в таблице 2.4.

Таблица 2.4 – Атрибуты сущности «Производители»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_firmi	int(10) unsigned	NO	PRI	auto_increm.	Код производителя
Nazvanie	varchar(45)	NO			Название

В таблице 2.5 представлены атрибуты сущности «Характеристики типов оборудования».

Таблица 2.5 – Атрибуты сущности «Характеристики типов оборудования»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_harakteristiki	int(10) unsigned	NO	PRI	auto_increm.	Код характеристики
nazvanie	varchar(45)	NO			Название
id_tipa_complect	int(10) unsigned	NO		0	Код типа оборудования

В таблице 2.6 представлены атрибуты сущности «Характеристики оборудования».

Таблица 2.7 – Атрибуты сущности «Характеристики оборудования»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_harakter_compl	int(10) unsigned	NO	PRI	auto_increm.	Код характеристики оборудования
id_complect	int(10) unsigned	YES	MUL		Код оборудования
id_harakteristiki	int(10) unsigned	NO	MUL	0	Код характеристики
znachenie	varchar(45)	NO			Значение
id_tipa_complect	int(10) unsigned	NO	MUL	0	Код типа оборудования
id	int(10) unsigned	NO	MUL	0	Код для связи

В таблице 2.8 представлены атрибуты сущности «Характеристики программного обеспечения».

Таблица 2.8 – Атрибуты сущности «Характеристики программного обеспечения»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_harakt_PO	int(10) unsigned	NO	PRI	auto_increm.	Код характеристики программного обеспечения
id_tipa_PO	int(10) unsigned	NO	MUL	0	Код типа программного обеспечения

id_PO	int(10) unsigned	NO	MUL	0	Код программного обеспечения
data_ustanovki	date	YES			Дата установки
data_posled_obnovl	date	YES			Дата последнего обновления
id_oborud	int(10) unsigned	YES	MUL		Код компьютера
data_neobh_obnovl	date	YES			Обновить (дата)

Сущность «Компьютеры» представлен IP-адрес компьютера в сети, наименованием отдела, сотрудником, тип компьютера. Атрибуты сущности «Компьютеры» приведены в таблице 2.9.

Таблица 2.9 – Атрибуты сущности «Компьютеры»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_oborud	int(10) unsigned	NO	PRI	auto_increm.	Код компьютера
id_tipa_oborud	int(10) unsigned	YES	MUL		Код типа компьютера
id_sotrud	int(10) unsigned	YES	MUL		Код сотрудника
id_otdela	int(10) unsigned	YES	MUL		Код отдела
imya_oborud	varchar(20)	NO			Имя компьютера
ip_adres	varchar(20)	NO			IP-адрес

В таблице 2.10 представлены атрибуты сущности «Обсуждение заявки».

Таблица 2.10 – Атрибуты сущности «Обсуждение заявки»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_obsujd	int(10) unsigned	NO	PRI	auto_increm.	Код сообщения
id_zayavki	int(10) unsigned	NO	MUL	0	Код заявки
data_vremya	datetime	YES			Дата и время
kommentarij	varchar(600)	YES			Комментарий
id_kem_oversh	int(10) unsigned	YES	MUL		Кто оставил
id_komu	int(10) unsigned	YES	MUL		Кому

В таблице 2.11 представлены атрибуты сущности «Отделы».

Таблица 2.11 – Атрибуты сущности «Отделы»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_otdela	int(10) unsigned	NO	PRI	auto_increm.	Код отдела
nazvanie	varchar(65)	NO			Название

В таблице 2.12 представлены атрибуты сущности «Программное обеспечение».

Таблица 2.12 – Атрибуты сущности «Программное обеспечение»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_PO	int(10) unsigned	NO	PRI	auto_increm.	Код программного обеспечения
id_tipa_PO	int(10) unsigned	NO	MUL	0	Тип программного обеспечения
nazvanie	varchar(45)	NO			Название
srok_dejstviya	int(10) unsigned	NO		0	Срок

В таблице 2.13 представлены атрибуты сущности «Склад».

Таблица 2.13 – Атрибуты сущности «Склад»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id	int(10) unsigned	NO	PRI	auto_increm.	Код склада
id_tipa_complect	int(10) unsigned	NO	MUL	0	Код типа оборудования
id_firmi	int(10) unsigned	YES	MUL		Код производителя
model	varchar(15)	YES			Модель
vsego	int(10) unsigned	YES			Всего

В таблице 2.14 представлены атрибуты сущности «Сотрудники».

Таблица 2.14 – Атрибуты сущности «Сотрудники»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_sotrud	int(10) unsigned	NO	PRI	auto_increm.	Код сотрудника
FIO	varchar(45)	NO			ФИО
id_otdela	int(10) unsigned	YES	MUL		Код отдела
id_doljnosti	int(10) unsigned	YES	MUL		Код должности
DR	date	YES			Дата рождения
Adres	varchar(75)	YES			Адрес
telefon	varchar(20)	YES			Телефон
e-mail	varchar(20)	YES			E-mail
kommentarij	varchar(200)	YES			Комментарий
rol	varchar(25)	YES			Роль
login	varchar(45)	YES			Логин
parol	varchar(45)	YES			Пароль

В таблице 2.15 представлены атрибуты сущности «Темы заявки».

Таблица 2.15 – Атрибуты сущности «Темы заявки»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_temi_zayavki	int(10) unsigned	NO	PRI	auto_increm.	Код темы заявки
nazvanie	varchar(45)	NO			Название

В таблице 2.16 представлены атрибуты сущности «Типы оборудования».

Таблица 2.16 – Атрибуты сущности «Типы оборудования»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_tipa_complect	int(10) unsigned	NO	PRI	auto_increm.	Код типа оборудования
nazvanie	varchar(45)	NO			Название

В таблице 2.17 представлены атрибуты сущности «Типы компьютеров».

Таблица 2.17 – Атрибуты сущности «Типы компьютеров»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_tipa_oborud	int(10) unsigned	NO	PRI	auto_increm.	Код типа компьютера
nazvanie	varchar(45)	NO			Название

В таблице 2.18 представлены атрибуты сущности «Типы программного обеспечения».

Таблица 2.18 – Атрибуты сущности «Типы программного обеспечения»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
Id_tipa_PO	int(10) unsigned	NO	PRI	auto_increm.	Код типа программного обеспечения
nazvanie	varchar(45)	NO			Название

В таблице 2.19 представлены атрибуты сущности «Заявки».

Таблица 2.19 – Атрибуты сущности «Заявки»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_zayavki	int(10) unsigned	NO	PRI	auto_increm.	Код заявки
id_sotrud	int(10) unsigned	NO	MUL	0	Код сотрудника
id_otdela	int(10) unsigned	NO	MUL	0	Код отдела

data	date	YES			Дата
data_zaversh	date	YES			Дата завершения
id_sotrud_teh	int(10) unsigned	YES	MUL		Код исполнителя
id_temi_zayavki	int(10) unsigned	YES	MUL		Код темы заявителя
vajnost	varchar(15)	NO			Важность
status	varchar(15)	NO			Статус
id_tipa_PO	int(10) unsigned	YES	MUL		Код типа ПО
id_PO	int(10) unsigned	YES	MUL		Название ПО

Описание обозначений базы данных представлено в приложении А.

2.1.4 Характеристика выходной информации для проектируемой системы

Результативная (выходная) информация – информация, формируемая на выходе системы.

Основными выходными документами проектируемой системы являются отчеты:

- «Отчет по заявкам»;
- «Отчет по состоянию склада»;
- «Отчет по срокам обновления ПО».

«Отчет по заявкам» должен отражать перечень зарегистрированных в системе заявок за определенный временной интервал. Реквизитами отчета служат такие поля как: номер заявки; дата заявки; ФИО сотрудника, создавшего заявку; ФИО исполнителя заявки, статус заявки, название программного обеспечения и тип, к которому оно относится, а также дата завершения заявки.

«Отчет по состоянию склада» должен предоставлять данные о количестве используемого и оставшегося оборудования каждого вида на складе. Реквизитами отчета служат поля: тип оборудования; производитель; модель; общее количество оборудования данного вида; количество

оборудования и использовании; количество оставшегося оборудования на складе.

«Отчет по срокам обновления ПО» – отчет, содержащий сведения о сроках обновления программных продуктов на каждом рабочем месте. Сведения в отчете должны быть упорядочены по возрастанию даты обновления ПО. Реквизитами данного документа могут быть следующие поля: ФИО сотрудника; отдел сотрудника; название ПО; тип ПО; дата обновления ПО.

2.2 Физическое моделирование информационной системы

Для реализации проекта нами была выбрана среда объектно-ориентированного программирования «Delphi 7».

Delphi – это среда разработки программ, ориентированных на работу в операционных системах семейства Windows. Программы в Delphi создаются на основе современной технологии визуального проектирования [22].

Разработанная нами информационная система состоит из 40 форм, перечень которых представлен в приложении Б.

Основные функциональные возможности системы представлены на рисунке 2.6 в виде дерева функций.

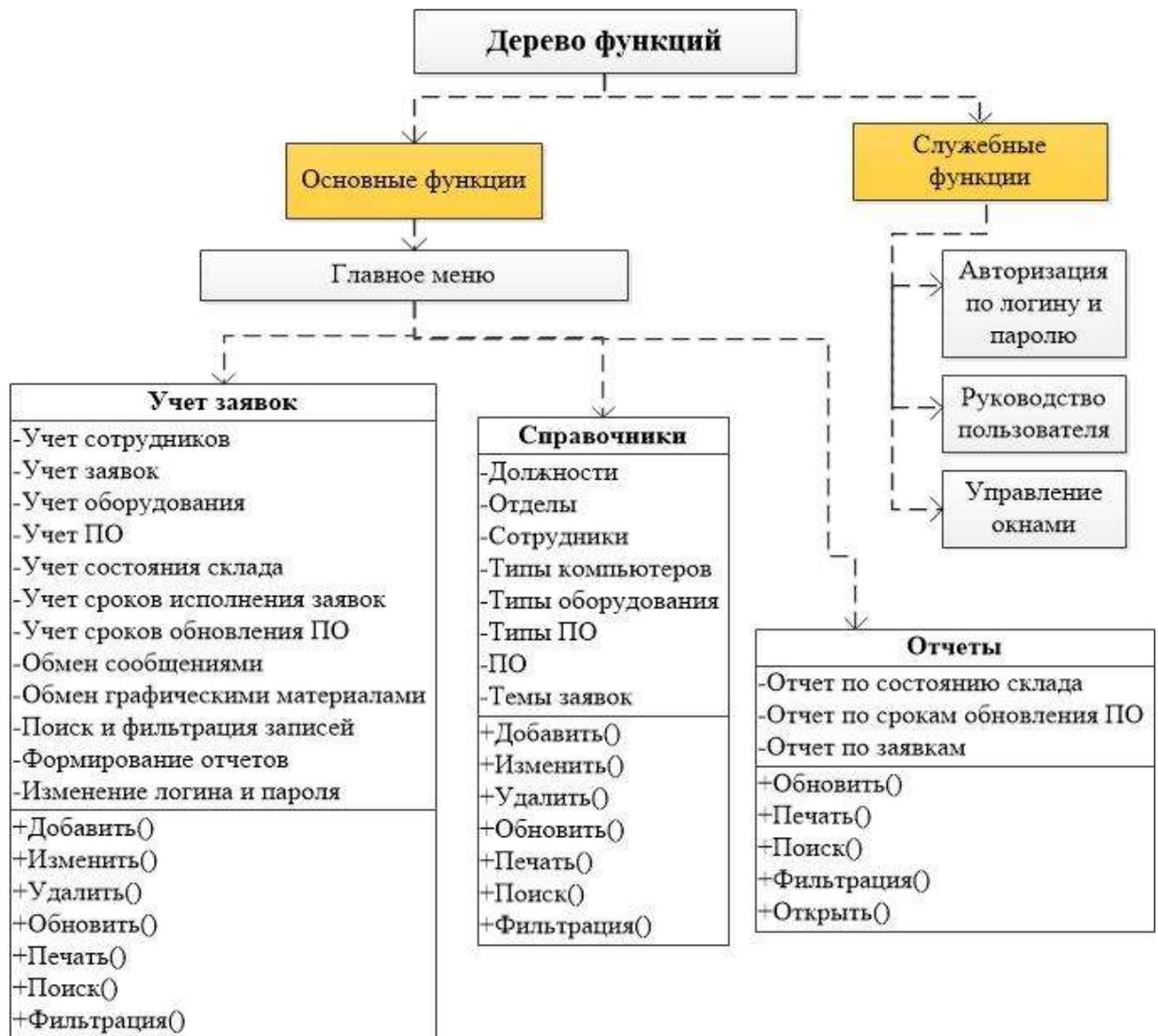


Рисунок 2.6 – Дерево функций

2.3 Описание программных модулей

После успешной авторизации в системе рисунок 2.7, пользователь, в зависимости от своей должности, получит доступ к одному из двух модулей программы: «Системный администратор» или «Пользователи».



Рисунок 2.7 – Форма «Вход в систему»

Рассмотрим особенности работы с каждым модулем.

Главное меню модуля «Администратор» имеет следующую структуру представленная на рисунке 2.8.

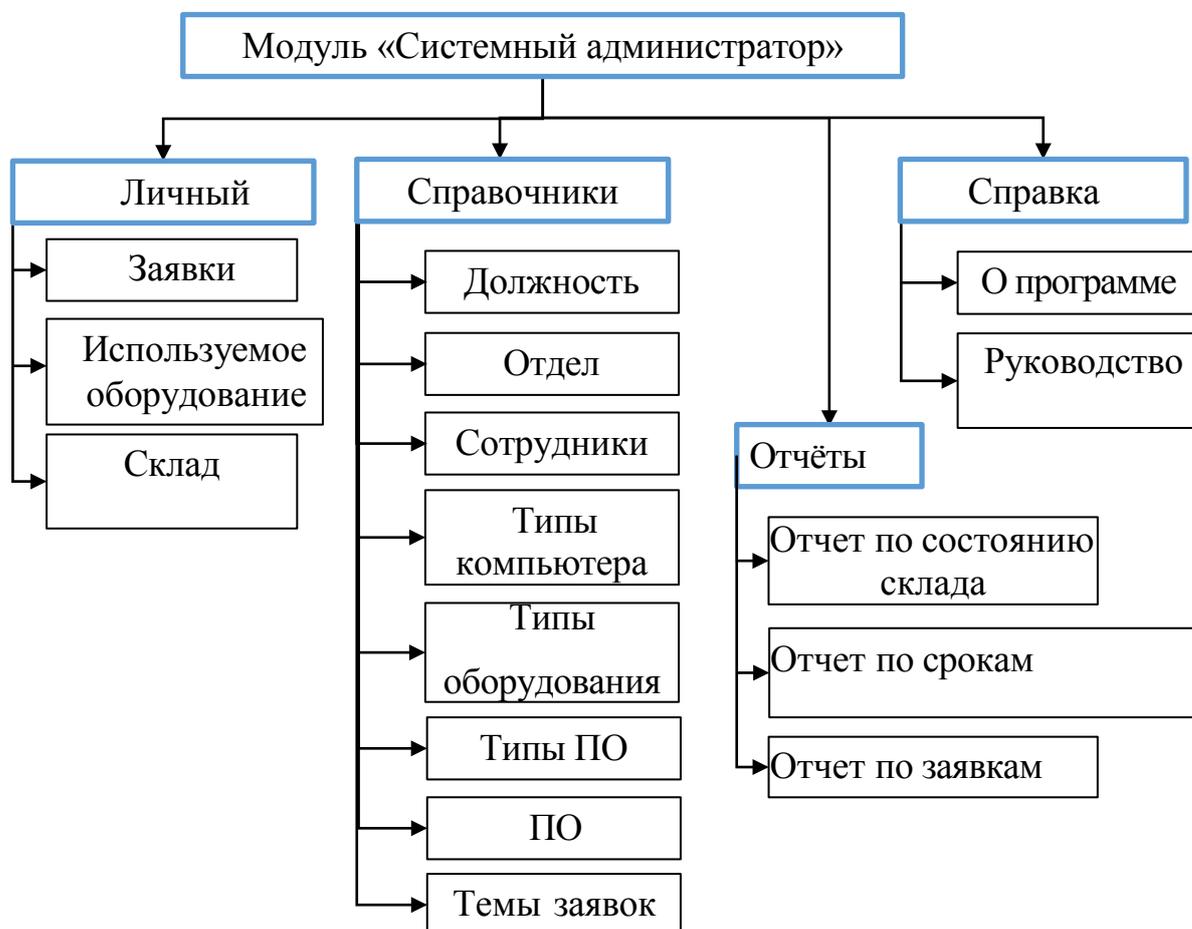


Рисунок 2.8 – Структура меню модуля «Системный администратор»

Раздел меню «Справочники» содержит перечень всех справочников системы. Доступ к добавлению, редактированию или удалению данных в справочниках имеют только пользователи с ролью системного администратора. Справочник «Должности» содержит перечень должностей сотрудников компании. Ввод названия должности осуществляется на форме, представленной на рисунке 2.9.

Ввод названия должности

Должность

Системный администратор

Сохранить Закрыть

Рисунок 2.9 – Форма «Ввод названия должности»

Справочник «Отделы» хранит перечень структурных подразделений компании. Ввод названия отдела производится на форме «Ввод названия отдела» представлено на рисунке 2.10.

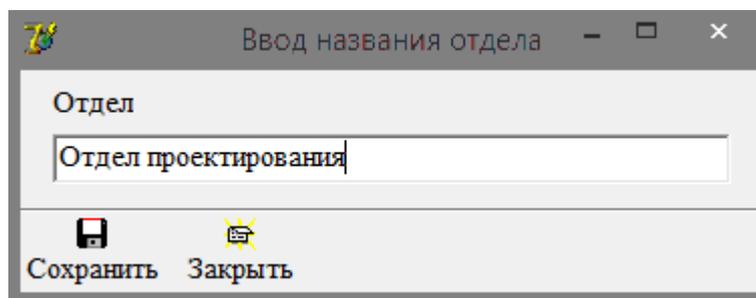


Рисунок 2.10 – Ввод названия отдела

Справочник «Сотрудники» позволяет вести учет сотрудников компании. Форме «Ввод данных о сотруднике» представлена на рисунке 2.11.

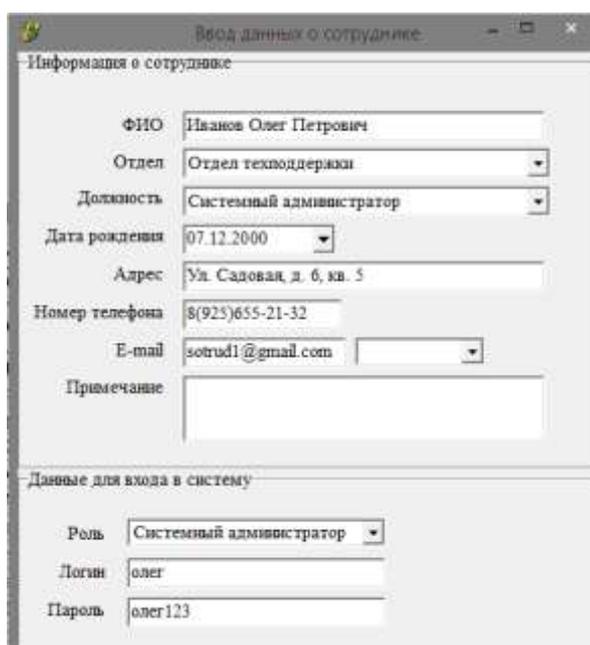


Рисунок 2.11 – Форма «Ввод данных о сотруднике»

Справочник «Типы компьютеров» позволяет вести список типов компьютеров, используемых в компании. Ввод названия типа компьютера производится на форме, представленной на рисунке 2.12.

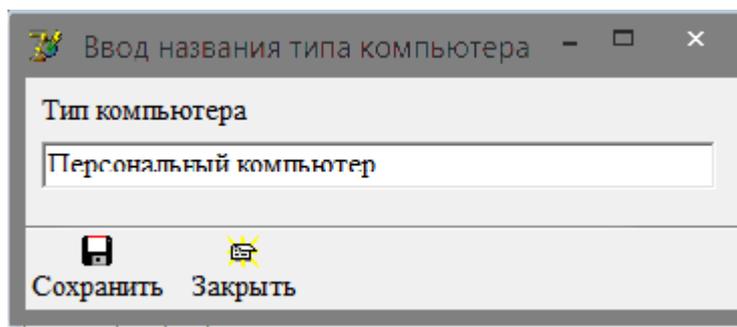


Рисунок 2.12 – Форма «Ввод названия типа компьютера»

Справочник «Типы оборудования» необходим для хранения перечня типов оборудования. Для ввода данных о типе оборудования используют форму «Ввод названия типа оборудования» представлено на рисунке 2.13.

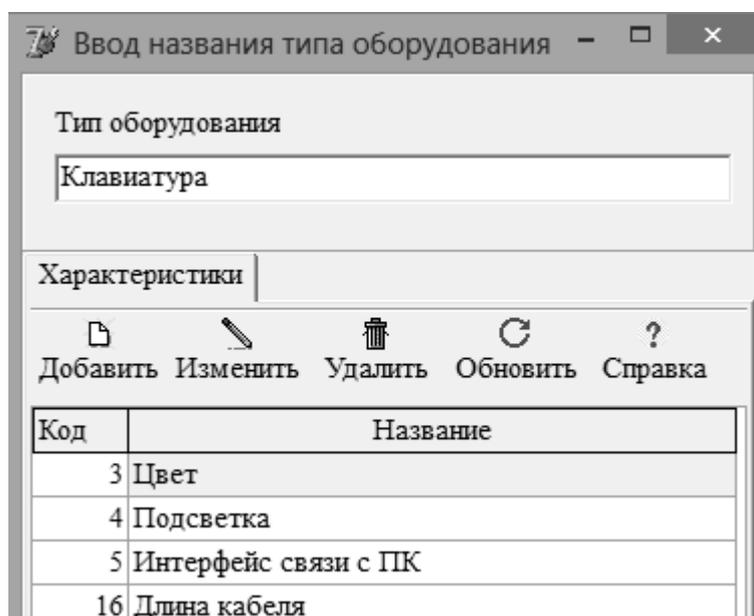


Рисунок 2.13 – Форма «Ввод названия типа оборудования»

Для ввода характеристик оборудования – форму, представленную на рисунке 2.14.

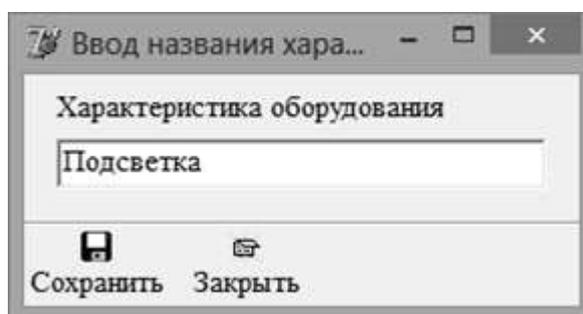


Рисунок 2.14 – Форма «Ввод названия характеристики оборудования»

Справочник «Типы ПО» позволяет вести учет типов используемого в компании программного обеспечения. Ввод названия типа программного

обеспечения производится на форме «Ввод названия типа ПО» представлено на рисунке 2.15.

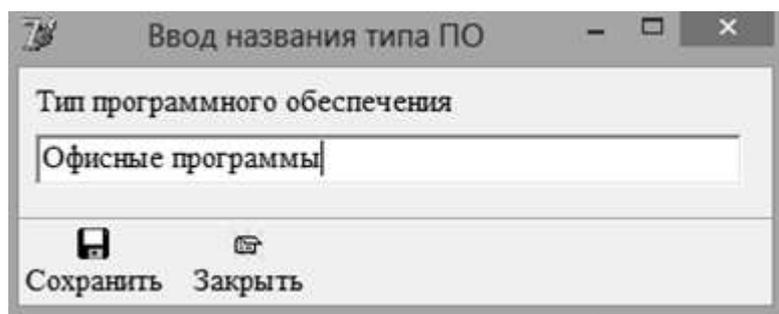


Рисунок 2.15 – Форма «Ввод названия типа ПО»

Справочник «Программное обеспечение» хранит перечень программ, используемых в компании. Для ввода данных о ПО предусмотрена форма, представленная на рисунке 2.16.

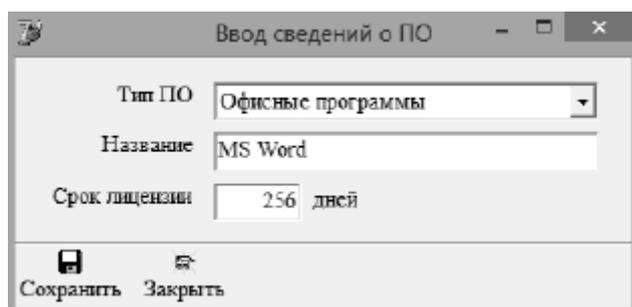


Рисунок 2.16 – Форма «Ввод сведений о ПО»

Справочник «Темы заявок» содержит перечень тем для заявок. Ввод темы заявки производится на форме «Ввод темы заявки» представлено на рисунке 2.17.

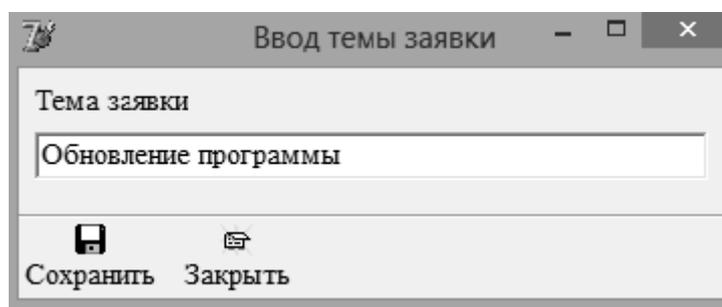


Рисунок 2.17 – Форма «Ввод темы заявки»

Пункт меню «Заявки» позволяет вести учет всех заявок, создаваемых сотрудниками компании представлено на рисунке 2.18.

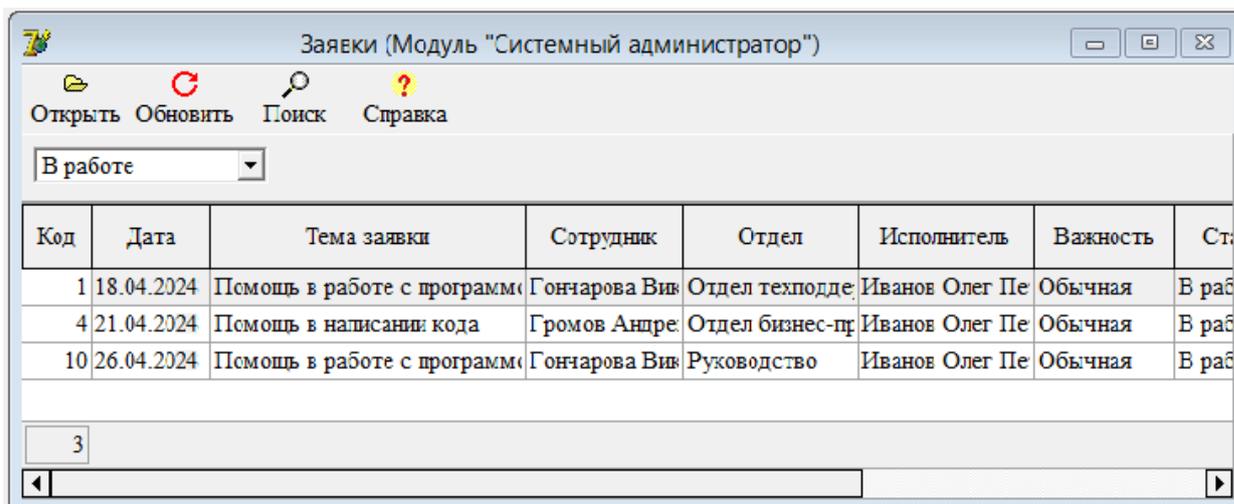


Рисунок 2.18 – Форма «Заявки (Модуль «Системный администратор»)»

Форма «Ввод данных о заявке» представлена на рисунке 2.19.

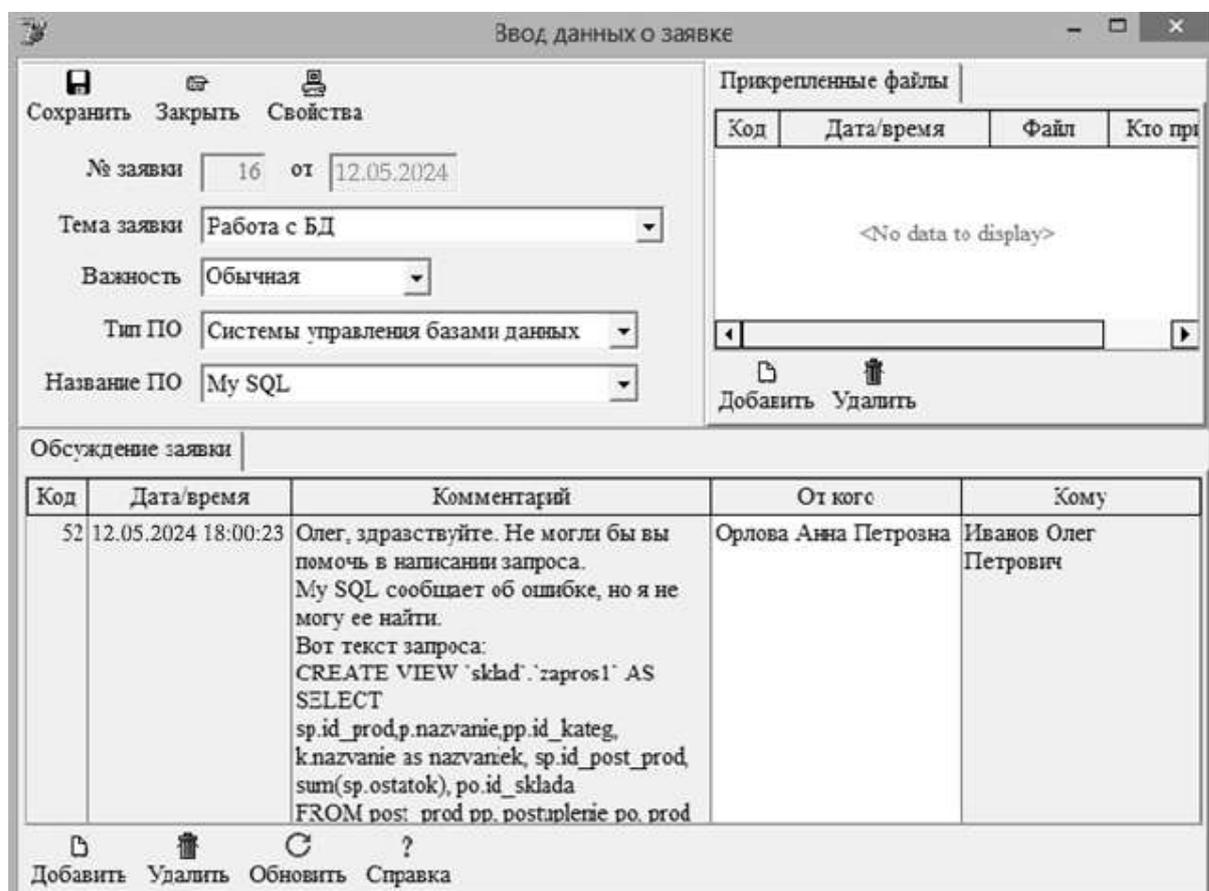


Рисунок 2.19 – Форма «Ввод данных о заявке»

Ввод нового сообщения производится на форме «Ввод сообщения» представлено на рисунке 2.20.

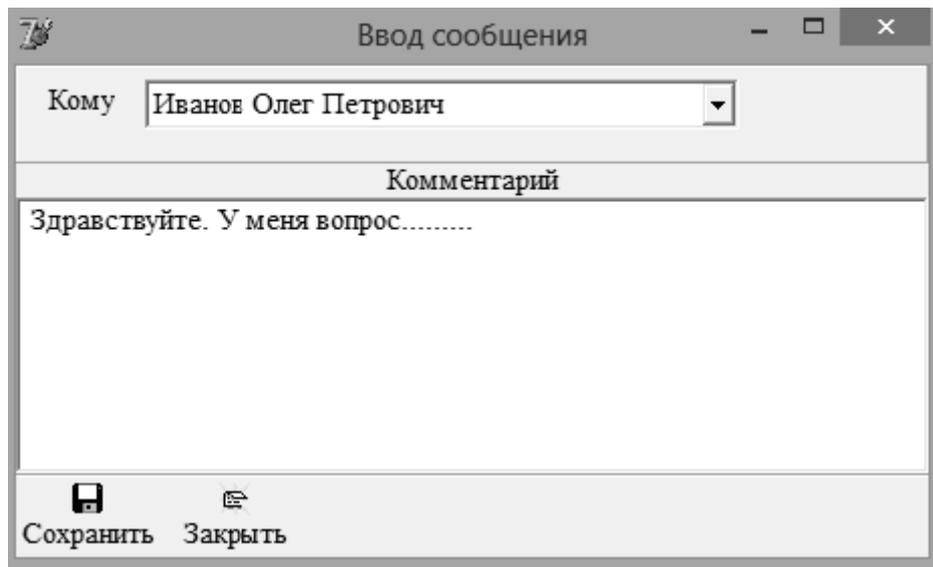


Рисунок 2.20 – Форма «Ввод сообщения»

В процессе исполнения заявок сотрудникам Отдел техподдержки необходим оперативный доступ к информации о состоянии компьютера и программного обеспечения, установленного на компьютере, обратившегося за помощью, сотрудника. Для этого необходимо перейти на форму «Ввод данных об используемом оборудовании» представлено на рисунке 2.21.

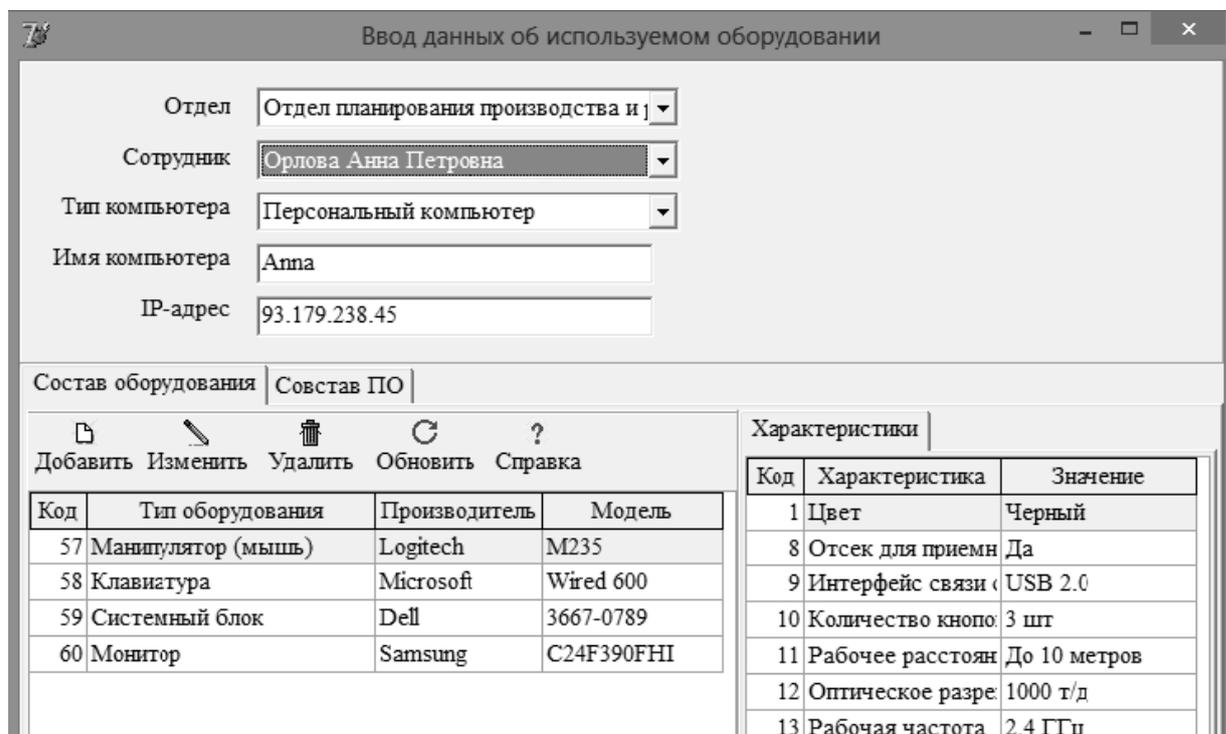


Рисунок 2.21 – Форма «Ввод данных об используемом оборудовании».

Вкладка «Состав оборудования» на рисунке 2.21 показывает, что система хранит сведения об имеющемся у сотрудника оборудовании и его

характеристиках, а также об установленном на компьютере ПО и его свойствах представлено на рисунке 2.22.

Код	Тип ПО	Название ПО	Дата установки	Дата последнего обновления	Дата следующего обновления
19	Офисные программы	MS Excel	29.01.2024	29.05.2024	12.06.2024
20	Офисные программы	MS Word	29.01.2024	29.05.2024	12.06.2024
21	Офисные программы	MS Power Point	29.01.2024	29.05.2024	12.06.2024
22	Офисные программы	MS Access	29.01.2024	29.05.2024	12.06.2024
23	Антивирусные программы	Kaspersky Antivirus	06.01.2024	06.05.2024	06.10.2024
24	Операционные системы	Microsoft Windows 7	06.01.2024	06.05.2024	20.05.2024

Рисунок 2.22 – Форма «Ввод данных об используемом оборудовании». Вкладка «Состав ПО»

Чтобы добавить новое оборудование, необходимо нажать кнопку «Добавить» на вкладке «Состав оборудования» и в открывшемся окне, где отражен перечень имеющегося на складе оборудования, выбрать нужную запись. После выделения какой-либо записи в данной таблице, в нижней части формы отразится остаток выбранного оборудования на складе представлено на рисунке 2.23.

Код	Тип оборудования	Производитель	Модель	Всего
1	Манипулятор (мышь)	Logitech	M235	15
2	Клавиатура	Microsoft	Wired 600	17
3	Системный блок	Dell	3667-0789	14
4	Манипулятор (мышь)	Lenovo	N3903	7
5	Монитор	Samsung	C24F390FHI	18

Остаток 4

Рисунок 2.23 – Форма «Выбор оборудования»

Для добавления записи о программном обеспечении, нужно перейти во вкладку «Состав ПО» формы «Ввод данных об используемом оборудовании» и нажать кнопку добавить. Ввод данных о программном обеспечении

производится на форме «Ввод данных о программном обеспечении» представлено на рисунке 2.24.

Рисунок 2.24 – Форма «Ввод данных о программном обеспечении»

Раздел меню «Отчеты» содержит 3 пункта, названия которых совпадают с именами формируемых в них отчетов.

Во всех отчетах системы можно проводить фильтрацию записей по интересующему пользователя временному интервалу. Для этого достаточно на верхней панели инструментов указать необходимый период и нажать кнопку «Отобразить».

Форма «Отчет о состоянии склада» содержит информацию о количестве каждого вида оборудования, находящегося в использовании и на складе представлено на рисунке 2.25.

Код	Тип оборудования	Производитель	Модель	Всего	В использовании	Остаток
1	Манипулятор (мышь)	Logitech	M235	15	9	6
2	Клавиатура	Microsoft	Wired 600	17	13	4
3	Системный блок	Dell	3667-0789	14	10	4
4	Манипулятор (мышь)	Lenovo	N3903	7	4	3
5	Монитор	Samsung	C24F390FHI	18	13	5

Рисунок 2.25 – Форма «Отчет о состоянии склада»

Форма «Отчет по срокам обновления ПО» представляет собой расписание обновления программного обеспечения, упорядоченное по дате представлено на рисунке 2.26.

Код	Тип ПО	Название ПО	Требуется обновить	Сотрудник	Отдел
25	Антивирусные програм	Kaspersky Antivirus	17.10.2024	Хрусталеv Константин Иван	Склад
8	Антивирусные програм	Kaspersky Antivirus	17.10.2024	Иванов Олег Петрович	IT-отдел
24	Операционные системы	Microsoft Windows 7	20.05.2025	Орлова Анна Петровна	Отдел планирования производст
15	Операционные системы	Microsoft Windows 80	12.06.2024	Петров Иван Петрович	Руководство
18	Офисные программы	MS Word	12.06.2024	Овсова Полина Сергеевна	Отдел кадров
19	Офисные программы	MS Excel	12.06.2024	Орлова Анна Петровна	Отдел планирования производст
20	Офисные программы	MS Word	12.06.2024	Орлова Анна Петровна	Отдел планирования производст
21	Офисные программы	MS Power Point	12.06.2024	Орлова Анна Петровна	Отдел планирования производст
22	Офисные программы	MS Access	12.06.2024	Орлова Анна Петровна	Отдел планирования производст
9	Операционные системы	Microsoft Windows 80	12.06.2024	Иванов Олег Петрович	IT-отдел
10	Системы автоматизации	BPWin	17.07.2024	Иванов Олег Петрович	IT-отдел

Рисунок 2.26 – Форма «Отчет по срокам обновления ПО»

Форма «Отчет по заявкам» хранит сведения обо всех заявках, созданных в системе за определенный период представлено на рисунке 2.27.

Код	Дата	Тема заявки	Сотрудник	Исполнитель	Статус	Название ПО	Дата завершения
1	18.04.2024	Помощь в работе с программ	Гончарова Вик	Иванов Олег Пе	В работе	MS Word	
2	20.04.2024	Помощь в работе с программ	Орлова Анна Г	Иванов Олег Пе	Завершена	MS Word	
3	21.05.2024	Обновление программы	Хрусталеv Ксв	Иванов Олег Пе	Завершена	Kaspersky Antiv	
4	21.05.2024	Помощь в написании кода	Громов Андре	Иванов Олег Пе	В работе	MS Access	
5	21.05.2024	Установка программы	Соколов Иван	Иванов Олег Пе	Завершена	Adobe Photosho	

Рисунок 2.27 – Форма «Отчет по заявкам»

Структура меню модуля «Пользователь» имеет следующий вид рисунок 2.28.

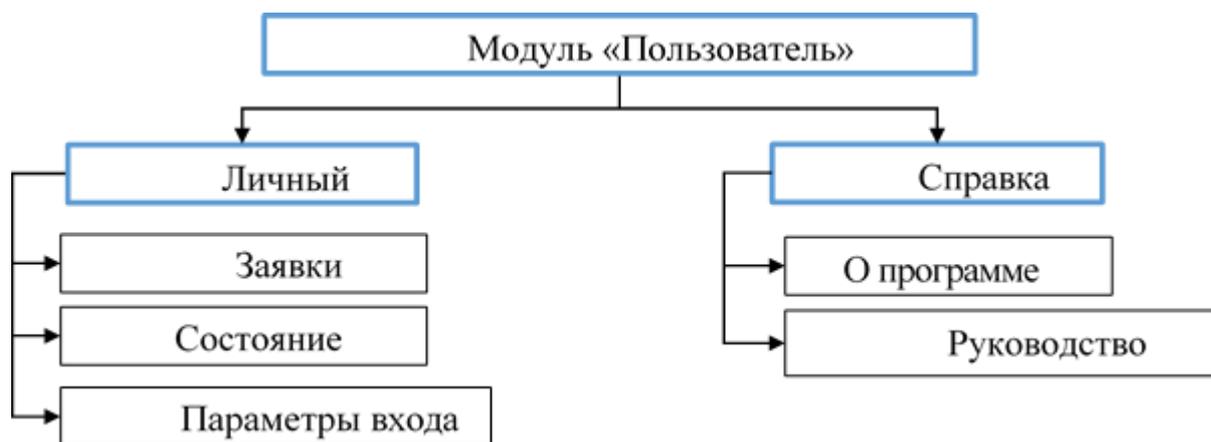


Рисунок 2.28 – Структура меню модуля «Пользователь»

Раздел меню «Личный кабинет» состоит из двух пунктов: «Заявки» и «Состояние компьютера» и «Параметры входа в систему».

Пункт меню «Заявки» позволяет создавать заявки на техническое обслуживание.

По умолчанию, при входе в систему, на экране отображаются заявки, имеющие статус «В работе». Для просмотра запросов с иным статусом, необходимо выбрать статус на верхней панели инструментов представлено на рисунке 2.29.

Код	Дата	Тема заявки	Сотрудник	Отдел	Исполнитель	Важность
2	20.05.2024	Помощь в работе с программ	Орлова Анна Г	Отдел планиров:	Иванов Олег Пе	Обычная
6	21.05.2024	Помощь в работе с программ	Орлова Анна Г	Отдел планиров:	Иванов Олег Пе	Обычная

Рисунок 2.29 – Форма «Заявки (Модуль «Пользователь»)

Ввод данных о заявке производится на форме, представленной на рисунке 2.30.

Ввод данных о заявке

Сохранить Закрыть Свойства

Тема заявки: Помощь в работе с программой

Важность: Обычная

Тип ПО: Офисные программы

Название ПО: MS Word

Прикрепленные файлы

Код	Дата/время	Файл	Кто при
<No data to display>			

Добавить Удалить

Обсуждение заявки

Код	Дата/время	Комментарий	От кого	Кому
5	20.05.2024 14:10:08	Здравствуйте, Олег. Не подскажете как создавать автосодержание?	Орлова Анна Петровна	Иванов Олег Петрович
6	20.05.2024 16:30:20	Здравствуйте, Анна. Возможно Вам поможет этот видеурок https://www.youtube.com/watch?v=VhL-vPtKbdE	Иванов Олег Петрович	
16	21.05.2024 12:34:10	Спасибо. Разобралась.	Орлова Анна Петровна	Иванов Олег Петрович

Добавить Обновить Справка

Рисунок 2.30 – Форма «Ввод данных о заявке»

Процедура ввода сообщения и прикрепления файлов аналогична той, что описана в модуле «Системный администратор» при работе с данной формой.

В пункте меню «Параметры входа в систему» пользователь может изменить логин и пароль для входа в систему.

ЗАКЛЮЧЕНИЕ

Для успешного и прибыльного функционирования любой современной компании в условиях растущей конкуренции и требований бизнеса к качеству и скорости выполняемых операций на компьютере, необходимо внедрение информационных систем автоматизированного учета обслуживания материально-технической базы.

Моделирование существующей технологии учета заявок на предприятии позволило выявить проблемы, с которыми сталкиваются сотрудники отдела техподдержки в процессе выполнения своих обязанностей.

В целях поиска решения, способного устранить выявленные проблемы учета, был проведен анализ ИС от известных разработчиков, который показал, что наиболее рациональным решением для исследуемой компании является разработка собственной системы.

Модель предлагаемой технологии учета заявок (т.е. после внедрения собственной ИС) должна устранить все существовавшие ранее недостатки учета.

Для разработки системы была выбрана среда программирования «Delphi7», так как данный язык является основным на предприятии, и СУБД «MySQL», так как компания уже много лет использует его.

Разработанная система состоит из двух модулей: «Системный администратор» и «Пользователь». Доступ к каждому модулю осуществляется посредством ввода логина и пароля.

Программа обеспечивает решение следующих задач: регистрация, редактирование и удаление заявок; прием, обработка и исполнение заявок; обмен сообщениями и графическими материалами; подготовка рабочего места сотрудника; учет сотрудников; добавление новых пользователей в систему; учет используемого оборудования и их характеристик; учет программного обеспечения, установленного на каждом компьютере;

контроль состояния склада технических средств; формирование и вывод на печать отчетов; смена логина и пароля.

Разработанный программный продукт достаточно недорогой и простой в использовании.

Таким образом, задачи решены в полном объеме, цель настоящей ВКР достигнута.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Агальцов, В.П. Базы данных. В 2-х т. Т. 2. Распределенные и удаленные базы данных: Учебник / В.П. Агальцов. – М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2019. – 272 с.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений, 3-е изд. / Г. Буч и др. // Пер. с англ. – М.: «И.Д. Вильямс», 2020. – 720 с.
3. Вдовин В.М. Предметно-ориентированные экономические информационные системы [Электронный ресурс]: учебное пособие / В. М. Вдовин, Л. Е. Суркова, А. А. Шурупов. – 3-е изд. – Москва: Дашков и К°, 2021. – 388 с.: ил. – ISBN 978-5-394-02262-3.
4. Голицына, О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. – М.: Форум, 2022. – 400 с.
5. Гольцман В. MySQL 5.0. Библиотека программиста: Питер; Санкт-Петербург; 2019.
6. Диго С.М. Базы данных: проектирование и использование: уч-к.– Москва: Финансы и статистика, 2019. – 595С.
7. Желонкин А. «Основы программирования в интегрированной среде Borland Delphi ». 2021г. – 240 с.
8. Избачков Ю.С. Информационные системы: учебник для вузов / Ю.С. Избачков, В.Н. Петров. – 2-е изд.– Санкт-Петербург: Питер, 2015. – 656 с.
9. Карпова, И.П. Базы данных: Учебное пособие / И.П. Карпова. – СПб.: Питер, 2021. – 240 с.
10. Кириллов В.В. Введение в реляционные базы данных. Введение в реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. – СПб.: БХВ-Петербург, 2020. – 464 с.
11. Кэнту М. Borland Delphi 7: Для профессионалов. – Санкт-Петербург: Питер, 2021.

12. Пирогов В.Ю. Информационные системы и базы данных: организация и проектирование: Учебное пособие / В.Ю. Пирогов. – СПб.: БХВ–Петербург, 2022. – 528 с.

13. Шеер Август–Вильгельм. Моделирование бизнес – процессов. М.: Весть– МетаТехнология, 2020. –175 С.

14. АО «ВОЕНТЕЛЕКОМ» [Электронный ресурс] // URL: <https://voentelecom.ru/> (дата обращения: 01.04.2024)

15. Система «1С: ServiceDesk» [Электронный ресурс]. URL: <http://www.audit-escort.ru/programms/servicedesk-sluzhba-podderzhki/#tab1-css> (дата обращения: 25.04.2024).

16. Система «HelpDesk» [Электронный ресурс]. URL: <https://it-guild.com/services/implementation/helpdesk/> (дата обращения: 25.04.2024).

17. Система «ServiceNow» [Электронный ресурс]. URL: <https://itsm365.ru/tour/> (дата обращения: 25.04.2024).

18. Система «WorkFlowSoft» [Электронный ресурс]. URL: <https://workflowsoft.ru/requests> (дата обращения: 25.04.2024)

19. Steven Alter. Information systems: Fundamentals of e-business, Publishing JohnWileyandSons, 2020

20. Laudon K., Laudon D. Management Information systems. Publishing EBSCO, 2019

Описание обозначений базы данных

Primary key (PRI) – первичный ключ.

Foreign key (MUL) – внешний ключ.

Unsigned – данное свойство означает, что в столбце запрещены отрицательные значения.

Auto_increment – данное свойство обеспечивает автоматическую нумерацию строк таблицы.

Описание типов данных:

IntegerНННInt – целое число в диапазоне либо от -2 147 483 648 до 2 147 483 647, либо (если указано свойство Unsigned) от 0 до 4 294 967 295.

Date– lara в формате «уууу-мм-дд», в диапазоне от «0000-01-01» до «9999-12-31».

Varchar – символьная строка переменной длины, содержащая не более указанного количества символов (от 0 до 65 535).

DecimalНННDec – точное (неокругляемое) число с фиксированной точкой. Может содержать до 65 значащих цифр и до 30 цифр после десятичного разделителя (в нашем случае - 8 значащих цифр и 2 после десятичного разделителя).

ПРИЛОЖЕНИЕ Б. Листинг программы

```
unit Unit1;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons;
type
TForm1 = class(TForm)
Label1: TLabel;
Label2: TLabel;
Edit1: TEdit;
Edit2: TEdit;
BitBtn2: TBitBtn;
Edit5: TEdit;
Edit6: TEdit;
Button1: TButton;
Button2: TButton;
Button3: TButton;
Button4: TButton;
Edit3: TEdit;
procedure Bit-
Btn2Click(Sender: TObject);
procedure But- ton1Click(Sender: TObject);
procedure But- ton2Click(Sender: TObject);
procedure But- ton3Click(Sender: TObject);
procedure But- ton4Click(Sender: TObject);
private
{ Private declarations } public
{ Public declarations }
end;
var
Form1: TForm1;
implementation
uses Unit5, Unit4, Unit3, Unit17, Unit27;
{$R *.dfm}
procedure
TForm1.BitBtn2Click(Sender: TObject);
begin
if (Edit1.Text=EmptyStr) or (Ed- it2.Text=EmptyStr) then begin ShowMes- sage('BBe,anTenornHnnaponb');
end
else begin
dm.t44.Close;
dm.t44.SQL.Clear;
dm.t44.SQL.Add('Select * from sotrudniki where login LIKE "' +Edit1.Text+" and parol LIKE "' +Edit2.Text+"');
dm.t44.Open;
if dm.t44.RecordCount=0 then begin ShowMessage('HeBepHbinnornHnnnna роль. Попробуйтеещераз.');
```

```

if Edit6.Text='Администратор' then begin Button2.Click;
end
else if Edit6.Text='Менеджер' then begin Button3.Click;
end
else if
Edit6.Text='На4arbHHKCKHafla' then begin Button4.Click;
end
end;
procedure
 TForm1.Button2Click(Sender: TObject);
begin
Form3.WindowState:=wsMaximized;
Form3.show;
Form1.Hide;
end;
procedure
 TForm1.Button3Click(Sender: TObject);
begin
Form27.WindowState:=wsMaximized;
Form27.Label1.Caption:=Edit5.Text;
Form27.Label2.Caption:=Edit3.Text;
Form27.show;
Form1.Hide;
end;
procedure
 TForm1.Button4Click(Sender: TObject);
begin
Form17.WindowState:=wsMaximized;
Form17.Label1.Caption:=Edit5.Text;
Form17.Label2.Caption:=Edit3.Text;
Form17.show;
Form1.Hide;
Form17.show;
end;
end.
unit Unit2;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, jpeg;
type
 TForm2 = class(TForm)
Image1: TImage;
Timer1: TTimer;
procedure Timer1Timer(Sender: TObject);
procedure Image1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form2: TForm2;
implementation
{$R *.dfm}
procedure
 TForm2.Timer1Timer(Sender:
TObject);
begin
Timer1.Enabled := false;

```

```

end;
unit Unit3;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Menus;
type
TForm3 = class(TForm)
ScrollBox1: TScrollBox;
MainMenu1: TMainMenu;
N2: TMenuItem;
N7: TMenuItem;
N15: TMenuItem;
N16: TMenuItem;
N8: TMenuItem;
N17: TMenuItem;
N12: TMenuItem;
N19: TMenuItem;
N20: TMenuItem;
N5: TMenuItem;
N6: TMenuItem;
N23: TMenuItem;
N24: TMenuItem;
N3: TMenuItem;
N9: TMenuItem;
N10: TMenuItem;
N1: TMenuItem;
procedure N1Click(Sender:
TObject);
procedure N8Click(Sender:
TObject);
procedure N7Click(Sender:
TObject);
procedure N15Click(Sender: TObject);
procedure N12Click(Sender: TObject);
procedure N20Click(Sender: TObject);
procedure FormClose(Sender:
TObject; var Action: TCloseAction);
procedure N9Click(Sender: TObject);
procedure N6Click(Sender:
TObject);
procedure N23Click(Sender: TObject);
procedure N24Click(Sender: TObject);
procedure N10Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations } end;
var
Form3: TForm3;
implementation
uses Unit4, Unit7, Unit9, Unit11,
Unit14, Unit16, Unit1, Unit33, Unit5,
Unit37, Unit38, Unit39, Unit34;
{$R *.dfm}
procedure
TForm3.N1Click(Sender: TObject);
begin
Form4.Parent: =F orm3 .ScrollBox 1;
Form4.show;
end;
procedure

```

```

TForm3.N8Click(Sender: TObject);
begin
Form7.Parent:=Form3.ScrollBox1;
Form7.show;
end;
procedure
TForm3.N7Click(Sender: TObject);
begin
Form9.Parent:=Form3.ScrollBox1;
Form9.show;
end;
procedure
TForm3.N15Click(Sender: TObject);
begin
Form11.Parent:=Form3.ScrollBox1;
Form11.show;
end;
procedure
TForm3.N12Click(Sender: TObject);
begin
Form14.Parent:=Form3.ScrollBox1;
Form14.show;
end;
procedure
TForm3.N20Click(Sender: TObject);
begin
Form16.Parent:=Form3.ScrollBox1;
Form16.show;
end;
procedure
TForm3.FormClose(Sender: TObject; var Action: TCloseAction);
Var X: word;
begin
X :=
MessageDlg('Bbifl,encTBHTenHOxoTHT эпокинуть программу?', MTConfirmation, [MBYes, MBNo], 0);
If X = mrYes then Form1.close
else Form1.Show;
end;
procedure
TForm3.N9Click(Sender: TObject);
begin
Form33.Parent:=Form3.ScrollBox1;
Form33.show;
end;
procedure TForm3.N6Click(Sender: TObject);
begin
dm.t18.Close;
dm.t18.SQL.Clear;
dm.t18.SQL.Add('SELECT * FROM zaprosi where status like "За-вершен");
dm.t18.Open;
Form37.Parent:=Form3.ScrollBox1;
Form37.show;
end;
procedure
TForm3.N23Click(Sender: TObject);
begin
dm.t32.Close;
dm.t32.Open;
Form38.Parent:=Form3.ScrollBox1;
Form38.show;
end;
procedure

```

```

TForm3.N24Click(Sender: TObject);
begin
Form39.Parent: =Form3.ScrollBox1;
Form39.show;
end;
procedure
TForm3.N10Click(Sender: TObject);
begin
Form34.WebBrowser1.Navigate( ExtractFilePath(Application.ExeName) + 'Помощь.htm');
Form34.show;
end;
end.
unit Unit4;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, cxGraphics, cxControls, cxLookAndFeels, cxLookAndFeelPainters,
cxStyles, dxSkinsCore, dxSkinscxPCPainter, cxCustomData, cxFilter,
cxData, cxDataStorage, cxEdit, DB, cxDBData, cxDBLookupComboBox,
StdCtrls, Buttons, cxGridLevel, cxGridCustomTableView, cxGridTableView,
cxGridDBTableView, cxClasses, cxGridCustomView, cxGrid, ExtCtrls,
ComCtrls, ToolWin, ImgList, dxDockControl, dxDockPanel;
type
TForm4 = class(TForm)
Panel2: TPanel;
cxGrid2: TcxGrid;
cxGrid2DBTableView1:
TcxGridDBTableView;
cxGrid2Level1:
TcxGridLevel;
cxGrid2DBTableView1id_kateg:
TcxGridDBColumn;
cxGrid2DBTableView1nazvanie:
TcxGridDBColumn;
ImageList1: TImageList;
ToolBar1: TToolBar;
ToolButton1: TToolButton;
ToolButton2: TToolButton;
ToolButton3: TToolButton;
ToolButton4: TToolButton;
ToolButton5: TToolButton;
ToolButton6: TToolButton;
ToolButton7: TToolButton;
ToolButton8: TToolButton;
dxDockPanel1: TdxDockPanel;
dxFloatDockSite1:
TdxFloatDockSite;
Edit1: TEdit;
procedure ToolButton1Click(Sender: TObject);
procedure ToolButton2Click(Sender: TObject);
procedure ToolButton3Click(Sender: TObject);
procedure
cxGrid2DBTableView1CellDbClick(
Sender: TcxCustomGridTableView;
ACellViewInfo:
TcxGridTableDataCellViewInfo;
AButton: TMouseButton;
AShift: TShiftState; var

```

```

AHandled: Boolean);
procedure ToolButton8Click(Sender: TObject);
procedure ToolButton6Click(Sender: TObject);
procedure Edit1Change(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure ToolButton7Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form4: TForm4;
implementation
uses Unit5, Unit6, Unit34;
{$R *.dfm}
procedure TForm4.ToolButton1Click(Sender: TObject);
begin
dm.t1.Insert;
Form6.showModal;
end;
procedure TForm4.ToolButton2Click(Sender: TObject);
begin
Form6.showModal;
end;
procedure TForm4.ToolButton3Click(Sender: TObject);
begin
dm.t1.Delete;
end;
procedure TForm4.cxGrid2DBTableView1CellIDblClick(
Sender: TcxCustomGridTableView;
ACellViewInfo: TcxGridTableDataCellViewInfo;
AButton: TMouseButton;
AShift: TShiftState; var
AHandled: Boolean);
begin
ToolButton2.Click;
end;
procedure TForm4.ToolButton8Click(Sender: TObject);
begin
Form4.Close;
end;
procedure TForm4.ToolButton6Click(Sender: TObject);

```

```

begin
dxDockPanel 1 .Show;
end;
procedure
TForm4.Edit1Change(Sender:
TObject);
begin
if Edit1 .Text=Emptystr then begin
dm.t1.Filtered :=false;
end
else begin
dm.t1.Close;
dm.t1.Filtered: =true;
dm.t1.Filter:='nazvanie like
'+Edit1.Text+'%';
dm.t1.Open;
end
end;
procedure
TForm4.FormShow(Sender: TObject);
begin
dxDockPanel 1 .Hide;
end;
procedure
TForm4 .ToolButton7 Click(Sender:
TObject);
begin
Form34.WebBrowser1 .Navigate( ExtractFilePath(Application.ExeName) + 'Помощь.Ыш');
Form34.show;
end;
end.
unit Unit6;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Mask,
DBCtrls, ImgList, ComCtrls, ToolWin;
type
TForm6 = class(TForm)
ToolBar1: TToolBar;
ToolButton1: TToolButton;
ToolButton3: TToolButton;
ToolButton4: TToolButton;
ToolButton6: TToolButton;
ImageList1: TImageList;
DBEdit1: TDBEdit;
Label1: TLabel;
procedure ToolButton1Click(Sender: TObject);
procedure ToolButton6Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure ToolButton3Click(Sender: TObject);
procedure ToolButton4Click(Sender: TObject);
private
{ Private declarations } public
{ Public declarations } end;
var
Form6: TForm6;
implementation
uses Unit5;
{$R *.dfm}
procedure
TForm6 .ToolButton1Click(Sender:

```

```

TObject);
begin
dm.t1.Edit;
dm.t1.Post;
Form6.Close;
end;
procedure
TForm6.ToolButton6Click(Sender:
TObject);
Var X: word;
begin
X :=
MessageDlg('CoxpaHHTbn3MeHeHna?',
MTConfirmation, [MBYes, MBNo], 0);
If X = mrYes then ToolBut- ton1.Click
else dm.t1.Cancel;
Form6.Close;
end;
procedure
TForm6.FormClose(Sender: TObject; var Action: TCloseAction);
begin
dm.t1.Cancel;
end;
procedure
TForm6.ToolButton3Click(Sender:
TObject);
begin
dm.t1.Prior;
end;
procedure
TForm6.ToolButton4Click(Sender:
TObject);
begin
dm.t1.Next;
end;
end.
unit Unit7;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, cxGraphics, cxCon- trols, cxLookAndFeels, cxLookAndFeelPainters,
cxStyles, dxSkinsCore, dxSkinscxPCPainter, cxCustomData, cxFilter,
cxData, cxDataStorage, cxEdit, DB, cxDBData, cxContainer,
cxGridCustomT ableView, cxGridTableView, cxGridDBTa- bleView, Buttons,
StdCtrls, cxTextEdit, cxMaskEdit, cxDropDownEdit, cxLookupEdit,
cxDBLookupEdit, cxD- BLookupComboBox, cxGridLevel, cxClasses,
cxGridCustomView, cxGrid, ExtCtrls, ComCtrls, ToolWin, ImgList, DBCtrls,
Mask, dxDockControl, dxDockPanel;
type
TForm7 = class(TForm)
Panel2: TPanel;
cxGrid2: TcxGrid;
cxGrid2DBTableView1:
TcxGridDBTableView;
cxGrid2Level1:
TcxGridLevel;
cxGrid2DBTableView1id:
TcxGridDBCcolumn;
cxGrid2DBTableView1id_kateg: TcxGridDBCcolumn;
cxGrid2DBTableView1nazvanie:
TcxGridDBCcolumn;
cxGrid2DBTableView1cena_ed: TcxGridDBCcolumn;

```

```

ImageList1: TImageList;
ToolBar1: TToolBar;
ToolButton1: TToolButton;
ToolButton2: TToolButton;
ToolButton3: TToolButton;
ToolButton4: TToolButton;
ToolButton5: TToolButton;
ToolButton6: TToolButton;
ToolButton7: TToolButton;
ToolButton8: TToolButton;
Panel5: TPanel;
Label1: TLabel;
Button1: TButton;
DBEdit1: TDBEdit;
DBLookupComboBox1:
TDBLookupComboBox;
dxDockPanel1: TdxDockPan- el;
dxFloatDockSite1:
TdxFloatDockSite;
Edit1: TEdit;
procedure ToolBut-
ton1Click(Sender: TObject);
procedure ToolBut-
ton2Click(Sender: TObject);
procedure ToolBut-
ton3Click(Sender: TObject);
procedure
cxGrid2DBTableView1CellDbClick(
Sender: TcxCus- tomGridView;
ACellViewInfo:
TcxGridTableDataCellViewInfo;
AButton: TMouseButton;
AShift: TShiftState; var AHandled: Boolean);
procedure But- ton1Click(Sender: TObject);
procedure cxDBLookupCom- boBox1PropertiesChange(Sender: TObject);
procedure DBLookupCom- boBox1CloseUp(Sender: TObject);
procedure ToolBut- ton8Click(Sender: TObject);
procedure Ed- it1Change(Sender: TObject);
procedure ToolBut- ton6Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure ToolBut- ton7Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form7: TForm7;
implementation
uses Unit5, Unit8, Unit34;
{$R *.dfm}
procedure
TForm7.ToolButton1Click(S ender:
TObject);
begin
dm.t2.Insert;
Form8 .showModal;
end;
procedure
TForm7.ToolButton2Click(Sender:
TObject);
begin

```

```

Form8 .showModal;
end;
procedure
TForm7 .ToolButton3 Click(Sender:
TObject);
begin
dm.t2.Delete;
end;
procedure
TForm7.cxGrid2DBTableView1CellID
blClick(
Sender: TcxCustomGridTableView;
ACellViewInfo:
TcxGridTableDataCellViewInfo;
AButton: TMouseButton;
AShift: TShiftState; var AHandled: Boolean);
begin
ToolButton2.Click;
end;
procedure
TForm7.Button1Click(Sender:
TObject);
begin
dm.t2.Close;
dm.t2.Filtered:=true;
dm.t2.Filter:='[id_kateg] =
'+DBEdit1.Text+'";
dm.t2.Open;
end;
procedure
TForm7.cxDBLookupComboBox1Pro
pertiesChange(Sender: TObject);
begin
Button1.Click;
end;
procedure
TForm7 .DBLookupComboBox 1Close
Up(Sender: TObject);
begin
Button1.Click;
end;
procedure
TForm7 .ToolButton8Click(Sender:
TObject);
begin
Form7.Close;
end;
procedure
TForm7 .Edit1 Change(Sender:
TObject);
begin
X :=
MessageDlg(Xo\paaHTbH3\icieiH>i?',
MTConfirmation, [MBYes, MBNo], procedure
0); TForm15.ToolButton4Click(Sender:
If X = mrYes then ToolBut- TObject);
tonl.Click begin
else dm.t6.Cancel; dm.t6.Next;
Form15.Close; end;
procedure end;
end.

```

```

TForm15.FormClose(Sender: TObject; var Action: TCloseAction);    unit Unit16;
begin dm.t6.Cancel;      interface
end;    uses
Windows, Messages, SysUtils,
procedure    Variants, Classes, Graphics, Controls,
TForm15.ToolButton8Click(Sender:    Forms,
TObject);    Dialogs, cxGraphics, cxCon-
begin    trols, cxLookAndFeels,
dm.t6.Close;    cxLookAndFeelPainters,
dm.t6.Open;    cxContainer, cxEdit,
dm.t7.Close;    dxSkinsCore, ComCtrls, ToolWin,
dm.t7.Open;    ImgList, cxTextEdit,
end;    cxMaskEdit, cxDropDownEdit cxLookupEdit, cxDBLookupEdit,
procedure    cxDBLookupComboBox,
TForm15 .T oolButton 11 Click(Sender: TObject);    StdCtrls, Mask, DBCtrls;
begin    type
Form34.WebBrowser1.Navigate(    TForm16 = class(TForm)
ExtractF ilePath(Application.ExeName)    Label1: TLabel;
+ 'Помощь.htm');    Label2: TLabel;
Form34.show;    Label3: TLabel;
end;    Label4: TLabel;
Label5: TLabel;
procedure    Label6: TLabel;
TForm15 .T oolButton3 Click(Sender:    Label7: TLabel;
TObject);    Label8: TLabel;
begin    Label9: TLabel;
dm.t6.Prior;    Label10: TLabel;
end;    Label11: TLabel;
DBEdit1: TDBEdit;
DBEdit2: TDBEdit;

DBEdit3: TDBEdit;
DBEdit4: TDBEdit;
DBEdit5: TDBEdit;
DBEdit6: TDBEdit;
DBEdit7: TDBEdit;
DBEdit8: TDBEdit;
DBEdit9: TDBEdit;
cxDBLookupComboBox1:
T cxDBLookupComboBox;
cxDBLookupComboBox2:
TcxDBLookupComboBox;
ImageList1: TImageList;
ToolBar1: TToolBar;
ToolButton1: TToolButton;
ToolButton6: TToolButton;
procedure ToolBut-
ton6Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure ToolBut-
ton1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form16: TForm16;
implementation
uses Unit5;
{$R *.dfm}
procedure

```

```

TForm16.ToolButton6Click(Sender: TObject);
Var X: word;
begin
X :=
MessageDlg('CoxpaHHTbH3MeHeHM?',
MTConfirmation, [MBYes, MBNo], 0);
If X = mrYes then ToolBut-
ton1.Click
else dm.t15.Cancel;
Form16.Close;
end;
procedure
TForm16.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
dm.t15.Cancel;
end;
procedure
TForm16.ToolButton1Click(Sender:
TObject);
begin
dm.t15.Edit;
dm.t15.Post;
Form16.Close;
end;
end.
unit Unit17;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Menus, StdCtrls, Ex- tCtrls;
type
TForm17 = class(TForm)
ScrollBox1: TScrollBox;
MainMenu1: TMainMenu;
N2: TMenuItem;
N1: TMenuItem;
N8: TMenuItem;
N7: TMenuItem;
N3: TMenuItem;
N9: TMenuItem;
N10: TMenuItem;
ScrollBox2: TScrollBox;
Panell: TPanel;
Label1: TLabel;
Label2: TLabel;
N4: TMenuItem;
N5: TMenuItem;
procedure N8Click(Sender: TObject);
procedure N7Click(Sender: TObject);
procedure N1Click(Sender: TObject);
procedure N5Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure N9Click(Sender: TObject);
procedure N10Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form17: TForm17;

```

```

implementation
uses Unit18, Unit22, Unit25, Unit5, Unit32, Unit1, Unit33, Unit35, Unit36,
Unit34;
{$R *.dfm}
procedure
TForm17.N8Click(Sender: TObject);
begin
Form18.Parent:=Form17.ScrollB ox1;
Form18.show;
end;
procedure
TForm17.N7Click(Sender: TObject);
begin
dm.t16.Close;
dm.t16.SQL.Clear;
dm.t16.SQL.Add('SELECT * from zapros1 where id_sklada like "'+Form22.DBEdit1.text+'");
dm.t16.Open;
dm.t17.Close;
dm.t17.Open;
Form22.Parent:=Form17.ScrollB ox1;
Form22.show;
end;
procedure
TForm17.N1Click(Sender: TObject);
begin
Form25 .Parent:=F orm17.ScrollB ox1;
dm.t18.Close;
dm.t18.Open;
dm.t12.Close;
dm.t12.Open;
Form25.show;
end;
procedure
TForm17.N5Click(Sender: TObject);
begin
Form32.Parent:=Form17.ScrollB ox1;
dm.t44.Close;
dm.t44.SQL.Clear;
dm.t44.SQL.Add('Select * from sotrudniki where id_sotrud LIKE "%'+Label2.Caption+'%");
dm.t44.Open;
Form32.show;
end;
procedure
TForm17.FormClose(Sender: TObject; var Action: TCloseAction);
Var X: word;
begin
X :=
MessageDlg('Bbifl,ehcTBHTeabHOxoTHT эпокинутьпрограмму?', MTConfirmation, [MBYes, MBNo], 0);
If X = mrYes then Form1.close
else Form1.Show;
end;
procedure
TForm17.N9Click(Sender: TObject);
begin
Form3 3.Parent:=Form17.ScrollB ox1;
Form33.show;
end;
procedure
TForm17.N10Click(Sender: TObject);
begin
Form34.WebBrowser1.Navigate( ExtractF ilePath(Application.ExeName) + 'Помощь.htm');
Form34.show;

```

```

end;
end.
unit Unit18;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, cxGraphics, cxCon- trols, cxLookAndFeels, cxLookAndFeelPainters,
cxStyles, dxSkinsCore, dxSkinscxPCPainter, cxCustomData, cxFilter,
cxData, cxDataStorage, cxEdit, DB, cxDBData, ImgList, ComCtrls, ToolWin,
cxGridLevel, cxGridCustom- TableView, cxGridTableView, cxGridDBTableView,
cxClasses, cxGridCustomView, cxGrid, ExtCtrls, cxDBLookupCom- boBox;
type
TForm18 = class(TForm)
Panel2: TPanel;
cxGrid2: TcxGrid;
cxGrid2DBTableView1:
TcxGridDBTableView;
cxGrid2Level1:
TcxGridLevel;
ToolBar1: TToolBar;
ToolButton1: TToolButton;
ToolButton2: TToolButton;
ToolButton3: TToolButton;
ToolButton4: TToolButton;
ToolButton7: TToolButton;
ToolButton8: TToolButton;
ImageList1: TImageList;
cxGrid2DBTableView1id_postupl:
TcxGridDBColumn;
cxGrid2DBT ableView1data: TcxGridDBColumn;
{$R *.dfm}
cxGrid2DBT ableView1id_sklada: TcxGridDBColumn;
cxGrid2DBT ableView1 summa:
TcxGridDBColumn;
cxGrid2DBTableView1Column1:
TcxGridDBColumn;
procedure ToolBut-
ton1Click(Sender: TObject);
procedure ToolBut-
ton2Click(Sender: TObject);
procedure ToolBut-
ton3Click(Sender: TObject);
procedure
cxGrid2DBTableView1CellDbIcIck(
Sender: TcxCus-
tomGridTableView;
ACellViewInfo:
TcxGridTableDataCellViewInfo;
AButton: TMouseButton;
AShift: TShiftState; var
AHandled: Boolean);
procedure ToolBut-
ton8Click(Sender: TObject);
procedure ToolBut-
ton4Click(Sender: TObject);
procedure ToolBut-
ton7Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations } end;
var

```

```

Form18: TForm18;
implementation
uses Unit5, Unit19, Unit34;
procedure
TForm18.ToolButton1Click(Sender: TObject);
begin
dm.t8.Insert;
dm.t8.FieldName('data').Value:=DateToStr(Date);
Form19.showModal;
end;
procedure
TForm18.ToolButton2Click(Sender: TObject);
begin
Form19.showModal;
end;
procedure
TForm18.ToolButton3Click(Sender: TObject);
Var X: word;
begin
if dm.t9.RecordCount=0 then begin
X := MessageDlg('Вбi действительно хотите удалить текущую за-пись?', MTConfirmation, [MBYes, MBNo],
0);
If X = mrYes then
dm.t8.Delete;
end
else begin ShowMessage('Сна4ана удалите све-дения о поступивших товарах!');
end
end;
procedure
TForm18.cxGrid2DBTableView1CellDblClick(
Sender: TcxCustomGridTableView;
ACellViewInfo:
TcxGridTableDataCellViewInfo;
AButton: TMouseButton;
AShift: TShiftState; var AHandled: Boolean);
begin
ToolButton2.Click;
end;
procedure
TForm18.ToolButton8Click(Sender:
TObject);
begin
Form18.Close;
end;
procedure
TForm18.ToolButton4Click(Sender:
TObject);
begin
dm.t8.Close;
dm.t8.Open;
dm.t9.Close;
dm.t9.Open;
end;
procedure
TForm18.ToolButton7Click(Sender:
TObject);
begin
Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.Бiш');
Form34.show;
end;

```



```

TRvDataSetConnection;
RvDataSetConnection192:
TRvDataSetConnection;
RvDataSetConnection193:
TRvDataSetConnection;
RvDataSetConnection194:
TRvDataSetConnection;
RvDataSetConnection195:
TRvDataSetConnection;
RvDataSetConnection196:
TRvDataSetConnection;
RvDataSetConnection197: TRvDataSetConnection;
procedure ToolBut- ton2Click(Sender: TObject);
procedure ToolBut- ton5Click(Sender: TObject);
procedure ToolBut- ton7Click(Sender: TObject);
procedure ToolBut- ton1Click(Sender: TObject);
procedure
cxGrid1DBTableView1CellDbClick(
Sender: TcxCus- tomGridView;
ACellViewInfo:
TcxGridTableDataCellViewInfo;
AButton: TMouseButton;
AShift: TShiftState; var AHandled: Boolean);
procedure But- ton1Click(Sender: TObject);
procedure ToolBut-
ton9Click(Sender: TObject);
procedure ToolBut- ton6Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure ToolBut- ton8Click(Sender: TObject);
procedure ToolBut- ton11Click(Sender: TObject);
procedure ToolBut-
ton3Click(Sender: TObject);
procedure ToolBut-
ton4Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form19: TForm19;
implementation
uses Unit20, Unit21, Unit5,
Unit34;
{$R *.dfm}
procedure
TForm19.ToolButton2Click(Sender:
TObject);
begin
dm.t8.Edit;
dm.t8.Post;
dm.t9.Insert;
dm.t9.FieldName('kolichestvo
').Value:='1';
Form20.showModal;
end;
procedure
TForm19.ToolButton5Click(Sender:
TObject);
begin
Form20.showModal;
end;

```

```

procedure
TForm19 .T oolButton7 Click(Sender:
TObject);
Var X: word;
begin
if dm.t10.RecordCount=0 then begin
X := MessageDlg('Вби действи-тельно хотите удалить текущую за-пись?', MTConfirmation, [MBYes, MBNo],
0);
If X = mrYes then
dm.t9.Delete;
Button1 .Click;
end
else begin
ShowMessage('Сна4ааа удалите све-дения о свойствах поступивших то-варов из таблицы ниже');
end
end;
procedure
TForm19.ToolButton1Click(Sender: TObject);
begin
dm.t8.Edit;
dm.t8.Post;
Form19.Close;
end;
procedure
TForm19.cxGrid 1DBT ableView 1 Cell DbClick(
Sender: TcxCus- tomGridView;
ACellViewInfo:
TcxGridTableDataCellViewInfo; AButton: TMouseButton;
AShift: TShiftState; var AHandled: Boolean);
begin
ToolButton5 .Click;
end;
procedure
TForm29 .T oolButton5 Click(Sender:
TObject);
begin
Form30.Label8.Visible:=false;
Form3 0 .Edit2 .Visible:=false;
Form30.Label9.Visible:=false;
Form3 0 .Edit 1.Visible:=false;
dm.t2.Close;
dm.t2 .Filtered: =true;
dm.t2.Filter:=[id] =
"+Form30.DBEdit7.Text+"";
dm.t2.Open;
Form3 0 .showModal;
end;
procedure
TForm29 .T oolButton7 Click(Sender:
TObject);
Var X: word;
begin
X :=
MessageDlg('Bbifl,eucTBHTenHOxoTHT еудалитьтекущуююзапись?', MTCon- firmation, [MBYes, MBNo], 0);
If X = mrYes then
dm.t19.Delete;
end;
procedure
TForm29 .T oolButton 1 Click(Sender:
TObject);
begin
dm.t18.Edit;

```

```

dm.t18.Post;
Form29.Close;
end;
procedure
TForm29.cxGrid1DBTableView1Cell
DblClick(
Sender: TcxCus- tomGridTableView;
ACellViewInfo:
TcxGridTableDataCellViewInfo;
AButton: TMouseButton;
AShift: TShiftState; var AHandled: Boolean);
begin
ToolButton5 .Click;
end;
procedure
TForm29.N1Click(Sender: TObject);
begin
dm.t18.Edit;
dm.t18.Post;
end;
procedure
TForm29.ToolButton6Click(Sender:
TObject);
Var X: word;
begin
X :=
MessageDlg('CoxpaHHTbU3MeHeHua?',
MTConfirmation, [MBYes, MBNo], 0);
If X = mrYes then ToolBut- ton1.Click
else dm.t18.Cancel;
Form29.Close;
end;
procedure
TForm29.FormClose(Sender: TObject; var Action: TCloseAction);
begin
dm.t18.Cancel;
end;
procedure
TForm29.ToolButton8Click(Sender:
TObject);
begin
dm.t19.Refresh;
end;
procedure
TForm29.BitBtn1Click(Sender:
TObject);
begin
dm.t4.Insert;
Form12 .showModal;
end;
procedure
TForm29.BitBtn2Click(Sender:
TObject);
begin
Form11 .showModal;
end;
procedure
TForm29 .T oolButton9Click(Sender:
TObject);
begin
RvProj ect29.Execute;
end;

```

```

procedure
TForm29.ToolButton11Click(Sender:
TObject);
begin
Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.Ыш');
Form34.show;
end;
procedure
TForm29.ToolButton3Click(Sender:
TObject);
begin
dm.t18.Prior;
end;
procedure
TForm29.ToolButton4Click(Sender:
TObject);
begin
dm.t18.Next;
end;
end.
unit Unit30;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, cxGraphics, cxCon- trols, cxLookAndFeels, cxLookAndFeelPainters,
cxContainer, cxEdit, dxSkinsCore, ImgList, ComCtrls, ToolWin, DBCtrls,
cxTextEdit, cxMaskEdit, cxD- BEdit, StdCtrls, Mask, cxDropDownEdit,
cxLookupEdit, cxD- BLookupEdit, cxDBLookupCom- boBox, cxSpinEdit;
type
TForm30 = class(TForm)
ToolBar1: TToolBar;
ToolButton1: TToolButton;
ToolButton6: TToolButton;
ImageList1: TImageList;
Label1: TLabel;
Label2: TLabel;
Label4: TLabel;
DBLookupComboBox1:
TDBLookupComboBox;
DBLookupComboBox2:
TDBLookupComboBox;
cxDBSpinEdit1: TcxDB- SpinEdit;
Label8: TLabel;
Button3: TButton;
Label9: TLabel;
DBEdit6: TDBEdit;
Edit1: TEdit;
DBEdit7: TDBEdit;
DBEdit3: TDBEdit;
Edit2: TEdit;
procedure ToolBut-
ton1Click(Sender: TObject);
procedure But-
ton3Click(Sender: TObject);
procedure DBLookupCom- boBox2CloseUp(Sender: TObject);
procedure ToolBut- ton6Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
{ Private declarations }
public

```

```

{ Public declarations }
end;
var
Form30: TForm30;
implementation
uses Unit5, Unit29;
{$R *.dfm}
procedure
TForm30.ToolButton1Click(Sender:
TObject);
begin
dm.t19.Edit;
dm.t19.Post;
Button3.Click;
Form30.Close;
end;
procedure
TForm30.Button3Click(Sender: TObject);
var i,j,h:real;
begin
dm.t16.Close;
dm.t16.SQL.Clear;
dm.t16.SQL.Add('SELECT * from zapros1 where id_prod like "%'+DBEdit3.Text+'%" and id_sklada like
"%'+Form29.DBEdit6.Text+'%"');
dm.t16.Open;
dm.t22.Close;
dm.t22.Open;
if dm.t16.RecordCount=0 then begin
Edit1.Text:='0';
Edit2.Text :=' Отсутствует';
Label8.Visible:=true;
Edit2.Visible:=true;
Label9.Visible:=true;
Edit1.Visible:=true;
end
else if dm.t22.RecordCount=0 then begin
i:=strtofloat(dm.t16.FieldByName('sum(pp.ostatok)').Value);
j:=0;
h:=i-j;
Edit1.Text:=floattostr(h);
Edit2.Text:=floattostr(i);
Label8.Visible:=true;
Edit2.Visible:=true;
Label9.Visible:=true;
Edit1.Visible:=true;
end
else begin
i:=strtofloat(dm.t16.FieldByName('sum(pp.ostatok)').Value);
j:=strtofloat(dm.t22.FieldByName('sum(z.kolichestvo)').Value);
h:=i-j;
Edit1.Text:=floattostr(h);
Edit2.Text:=floattostr(i);
Label8.Visible:=true;
Label9.Visible:=true;
Label8.Visible:=true;
Edit2.Visible:=true;
Label9.Visible:=true;
Edit1.Visible:=true;
end
end;
procedure
TForm30.DBLookupComboBox2Close

```

```

Up(Sender: TObject);
begin
Button3.Click;
end;
procedure
TForm30.ToolButton6Click(Sender:
TObject);
Var X: word;
begin
X :=
MessageDlg('CoxpaHHTbH3MeHeHHa?',
MTConfirmation, [MBYes, MBNo], 0);
If X = mrYes then ToolBut- ton1.Click
else dm.t19.Cancel;
Form30.Close;
end;
procedure
TForm30.FormClose(Sender: TObject; var Action: TCloseAction);
begin
dm.t19.Cancel;
end;
end.
unit Unit31;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, cxGraphics, cxCon- trols, cxLookAndFeels, cxLookAndFeelPainters,
cxStyles, dxSkinsCore, dxSkinscxPCPainter, cxCustomData, cxFilter,
cxData, cxDataStorage, cxEdit, DB, cxDBData, cxDBLookupCom- boBox,
cxGridLevel, cxGridCustom- TableView, cxGridTableView, cxGridDBTableView,
cxClasses, cxGridCustomView, cxGrid, cxSpinEdit, cxButtonEdit, StdCtrls,
ExtCtrls, Menus, ComCtrls, ToolWin, ImgList;
type
TForm31 = class(TForm) cxGrid2: TcxGrid; cxGridDBTableView1:
TcxGridDBTableView;
cxGridLevel1: TcxGridLevel;
PopupMenu1: TPopupMenu;
N1: TMenuItem;
cxGridDBTa- bleView1id_svojtva: TcxGridDBColumn;
cxGridDBTa- bleView1id_yach_hran: TcxGridDB- Column;
cxGridDBTa- bleView1id_prod: TcxGridDBColumn;
cxGridDBTa- bleView1Column1: TcxGridDBColumn;
ImageList1: TImageList;
ToolBar1: TToolBar;
ToolButton1: TToolButton;
ToolButton3: TToolButton;
procedure N1Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure ToolBut- ton1Click(Sender: TObject);
procedure ToolBut- ton3Click(Sender: TObject);
procedure cxGridDBTa- bleView1CellDbClick( Sender: TcxCus-
tomGridTableView;
ACellViewInfo: TcxGridTableDataCellViewInfo; AButton: TMouseButton;
AShift: TShiftState; var AHandled: Boolean);
private
{ Private declarations }
public
{ Public declarations } end;
var
Form31: TForm31;
implementation
uses Unit5, Unit24;

```

```

{$R *.dfm}
procedure
TForm31 .N1Click(Sender: TObject);
begin
T oolButton1 .Click;
end;
procedure
TForm31.FormClose(Sender: TObject; var Action: TCloseAction);
begin
dm.t23.Cancel;
end;
procedure
TForm31 .ToolButton1Click(Sender: TObject);
begin
dm.t23.Insert;
Form24.showModal;
dm.t23.Edit;
dm.t23 .FieldByName('id_prod').
Val-
ue:=dm.t12.FieldByName('id_prod').V alue;
dm.t23 .FieldByName('id_kateg'). Val-
ue:=dm.t12.F ieldByName('id_kateg').V alue;
dm.t23 .FieldByName('id_zapros _prod').Value:=dm.t12.FieldByName('i d_zapros_prod').Value;
dm.t23 .FieldByName('id_post_p rod').Value:=dm.t24.FieldByName('id_ post_prod').Value;
dm.t23 .FieldByName('id_yach_h ran').Value:=dm.t24.FieldByName('id_ yach_hran').Value;
dm.t23.Post;
end;
procedure
TForm31 .ToolButton3Click(Sender: TObject);
begin
dm.t23.Cancel;
end;
procedure
TForm31 .cxGridDBT ableViewICellD blClick(
Sender: TcxCus- tomGridT ableView;
ACellViewInfo:
TcxGridTableDataCellViewInfo;
AButton: TMouseButton;
AShift: TShiftState; var AHandled: Boolean);
begin
T oolButton1 .Click;
end;
end.
unit Unit32;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Mask,
DBCtrls, ImgList, ComCtrls, ToolWin;
type
TForm32 = class(TForm)
Label1: TLabel;
Label2: TLabel;
DBEdit1: TDBEdit;
DBEdit2: TDBEdit;
ToolBar1: TToolBar;
ToolButton1: TToolButton;
ToolButton3: TToolButton;
ImageList1: TImageList;
procedure ToolBut- ton1Click(Sender: TObject);
procedure ToolBut- ton3Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);

```

```

private
{ Private declarations } public
{ Public declarations } end;
var
Form32: TForm32;
implementation
uses Unit5;
{$R *.dfm}
procedure
TForm32.ToolButton1Click(Sender: TObject);
begin
dm.t44.Edit;
dm.t44.Post;
Form32.Close;
end;
procedure
TForm32.ToolButton3 Click(Sender: TObject);
Var X: word;
begin
X := MessageDlg('Сохранить изменения?', MTConfirmation, [MBYes, MBNo], 0);
If X = mrYes then ToolButton1.Click
else dm.t44.Cancel;
Form32.Close;
end;
procedure
TForm32.FormClose(Sender: TObject; var Action: TCloseAction);
begin
dm.t44.Cancel;
end;
end.
unit Unit35;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, cxGraphics, cxControls, cxLookAndFeels, cxLookAndFeelPainters,
cxStyles, dxSkinsCore, dxSkinscxPCPainter, cxCustomData, cxFilter,
cxData, cxDataStorage, cxEdit, DB, cxDBData, cxGridCustomTableview,
cxGridTableView, cxGridDBTableView, cxGridLevel, cxClasses,
cxGridCustomView, cxGrid, ComCtrls, ToolWin, ImgList, cxDBLookupComboBox,
StdCtrls, ExtCtrls, Buttons, cxContainer, cxTextEdit, cxMaskEdit, cxSpinEdit, cxDBEdit;
type
TForm35 = class(TForm)
Label1: TLabel;
Panel1: TPanel;
BitBtn1: TBitBtn;
cxDBSpinEdit1: TcxDBSpinEdit;
procedure BitBtn1Click(Sender: TObject);
private
{ Private declarations } public
{ Public declarations } end;
var
Form35: TForm35;
implementation
uses Unit5, Unit36, Unit17, Unit34;
{$R *.dfm}
procedure
TForm35.BitBtn1Click(Sender:
TObject);
var i,j,h: real;
begin

```

```

dm.t29.Edit;
dm.t29.FieldName('data_spisa niya').value:=DateToStr(Date);
dm.t29.Post;
i:=strtofloat(dm.t36.FieldName('ostatok').Value);
j:=strtofloat(dm.t29.FieldName('kolichestvo').Value);
h:=i-j;
dm.t36.Edit;
dm.t36.FieldName('ostatok').value:=floattostr(h);
dm.t36.Post;
dm.t16.Close;
dm.t16.Open;
Form35.Close;
end;
end.
unit Unit36;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, cxGraphics, cxControls, cxLookAndFeels, cxLookAndFeelPainters,
cxStyles, dxSkinsCore, dxSkinscxPCPainter, cxCustomData, cxFilter,
cxData, cxDataStorage, cxEdit, DB, cxDBData, cxGridCustomTableView,
cxGridTableView, cxGridDBTableView, ImgList, cxGridLevel, cxClasses,
cxGridCustomView, cxGrid, ComCtrls, ToolWin, cxDBLookupComboBox;
type
TForm36 = class(TForm)
ToolBar1: TToolBar;
ToolBar4: TToolBar;
ToolBar5: TToolBar;
ToolBar7: TToolBar;
ToolBar8: TToolBar;
cxGrid2: TcxGrid;
cxGrid2DBTableView1:
TcxGridDBTableView;
cxGrid2Level1: TcxGridLevel;
ImageList1: TImageList;
cxGrid2DBTableView1id_spis: TcxGridDBColumn;
cxGrid2DBTableView1id_yach_hran: TcxGridDBColumn;
cxGrid2DBTableView1ostatok: TcxGridDBColumn;
cxGrid2DBTableView1id_prod:
TcxGridDBColumn;
cxGrid2DBTableView1id_sklada:
TcxGridDBColumn;
cxGrid2DBTableView1Column1:
TcxGridDBColumn;
procedure ToolButton4Click(Sender: TObject);
procedure ToolButton8Click(Sender: TObject);
procedure ToolButton7Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form36: TForm36;
implementation
uses Unit5, Unit34;
{$R *.dfm}
procedure
TForm36.ToolButton4Click(Sender:

```

```

TObject);
begin
dm.t31.Close;
dm.t31.Open;
end;
procedure
TForm36.ToolButton8Click(Sender:
TObject);
begin
Form36.Close;
end;
procedure
TForm36.ToolButton7Click(Sender:
TObject);
begin
Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.Йш');
Form34.show;
end;
end.
unit Unit37;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, cxGraphics, cxCon- trols, cxLookAndFeels, cxLookAndFeelPainters,
cxStyles, dxSkinsCore, dxSkinscxPCPainter, cxCustomData, cxFilter,
cxData, cxDataStorage, cxEdit, DB, cxDBData, cxDBLookupCom- boBox, RpCon,
RpConDS, RpDefine, RpRave, Buttons, StdCtrls, Mask, DBCtrls, ComCtrls,
ExtCtrls, cxGridLevel, cxGridCustomTableView, cxGridTableView,
cxGridDBT ableView, cxClasses, cxGridCustomView, cxGrid, ImgList, ToolWin;
type
TForm37 = class(TForm) cxGrid2: TcxGrid; cxGridDBTableView1:
TcxGridDBTableView;
cxGridLevel1: T cxGridLevel;
Panel2: TPanel;
Label2: TLabel;
Label3: TLabel;
DateTimePicker1:
TDateTimePicker;
DateT imePicker2:
TDateTimePicker;
Button2: TButton;
RvProject30: TRvProject;
RvDataSetConnection301:
TRvDataSetConnection;
RvDataSetConnection302:
TRvDataSetConnection;
RvDataSetConnection303:
TRvDataSetConnection;
RvDataSetConnection304:
TRvDataSetConnection;
cxGridDBTa- bleView1id_zakaza: TcxGridDBCColumn;
cxGridDBTa-
bleView1data_dost: TcxGridDBCColumn;
cxGridDBTa-
bleView1id_menejera: TcxGridDB- Column;
cxGridDBTableView1status:
TcxGridDBColumn;
cxGridDBTa-
bleView1id_sklada: TcxGridDBCColumn;
cxGridDBTableView1data:

```

```

TcxGridDBColumn;
cxGridDBTableView1data_zaversh: TcxGridDB- Column;
ToolBar1: TToolBar;
ToolBar4: TToolBar;
ToolBar5: TToolBar;
ToolBar7: TToolBar;
ToolBar8: TToolBar;
ImageList1: TImageList;
ToolBar1: TToolBar;
ToolBar2: TToolBar;
ToolBar3: TToolBar;
procedure Button2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations } end;
var
Form37: TForm37;
implementation
uses Unit5, Unit29, Unit26, Unit34;
{$R *.dfm}
procedure
TForm37.Button2Click(Sender: TObject);
var StartDate, EndDate: string;
begin
StartDate:=formatDateTime('yyy y-mm-dd',DateTimePicker1.Date);
EndDate:=formatDateTime('yyy y-mm-dd',DateTimePicker2.Date);
dm.t18.Close;
dm.t18.SQL.Clear;
dm.t18.SQL.Text:='SELECT * FROM zaprosi z where (z.data between :StartDate AND :EndDate) and (status=
'Завершен')';
dm.t18.Parameters.ParamByName('StartDate').DataType:= ftDate;
dm.t18.Parameters.ParamByName('EndDate').DataType:= ftDate;
dm.t18.Parameters.ParamValues['StartDate']:=DateToStr(DateTimePicker1.Date);
dm.t18.Parameters.ParamValues['EndDate']:=DateToStr(DateTimePicker2.Date);
dm.t18.Open;
end;
procedure
TForm37.Button1Click(Sender: TObject);
begin
Form26.ShowModal;
end;
procedure
TForm37.Button4Click(Sender: TObject);
begin
dm.t18.Close;
dm.t18.Open;
end;
procedure
TForm37.Button8Click(Sender: TObject);
begin
Form37.Close;
end;
procedure
TForm37.Button7Click(Sender: TObject);
begin
Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)

```

```

+ 'Помощь.htm');
Form34.show;
end;
end.
unit Unit38;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, cxGraphics, cxCon- trols, cxLookAndFeels, cxLookAndFeelPainters,
cxStyles, dxSkinsCore, dxSkinscxPCPainter, cxCustomData, cxFilter,
cxData, cxDataStorage, cxEdit, DB, cxDBData, cxGridCustomTa- bleView,
cxGridT ableView, cxGridDBTableView, ImgList, ComCtrls, ToolWin, RpCon,
RpConDS, RpDefine, RpRave, StdCtrls, ExtCtrls, cxGridLevel, cxClasses,
cxGridCustomView, cxGrid, cxDBLookupComboBox;
type
TForm38 = class(TForm) cxGrid2: TcxGrid; cxGridDBTableView1:
TcxGridDBTableView;
cxGridLevel1: TcxGridLevel;
Panel2: TPanel;
Label2: TLabel;
Label3: TLabel;
DateTimePicker1:
TDateTimePicker;
DateTimePicker2:
TDateTimePicker;
Button2: TButton;
RvProject30: TRvProject;
RvDataSetConnection301:
TRvDataSetConnection;
RvDataSetConnection302: TRvDataSetConnection;
RvDataSetConnection303:
TRvDataSetConnection;
RvDataSetConnection304: TRvDataSetConnection;
ToolBar1: TToolBar;
ToolButton4: TToolButton;
ToolButton5: TToolButton;
ToolButton7: TToolButton;
ToolButton8: TToolButton;
ImageList1: TImageList;
cxGridDBTa-
bleView1id_prod: TcxGridDBCcolumn;
cxGridDBTa-
bleView1sumzpkolichestvo:
TcxGridDBCcolumn;
cxGridDBTa- bleView1id_kateg: TcxGridDBCcol- umn;
cxGridDBTa- bleView1id_sklada: TcxGridDBCcol- umn;
cxGridDBTableView1data: TcxGridDBCcolumn;
ToolButton1: TToolButton;
procedure But-
ton2Click(Sender: TObject);
procedure ToolBut- ton4Click(Sender: TObject);
procedure ToolBut- ton8Click(Sender: TObject);
procedure ToolBut- ton7Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations } end;
var
Form38: TForm38;
implementation
uses Unit5, Unit34;
{$R *.dfm}

```

```

procedure
TForm38.Button2Click(Sender: TObject);
var StartDate, EndDate: string;
begin
StartDate:=formatDateTime('yyy y-mm-dd',DateTimePicker1.Date);
EndDate:=formatDateTime('yyy y-mm-dd',DateTimePicker2.Date);
dm.t32.Close;
dm.t32.SQL.Clear;
dm.t32.SQL.Text:='SELECT * FROM zapros5 z where (z.data between :StartDate AND :EndDate)';
dm.t32.Parameters.ParamByName('StartDate').DataType:=ftDate;
dm.t32.Parameters.ParamByName('EndDate').DataType:=ftDate;
dm.t32.Parameters.ParamValues['StartDate']:=DateTimePicker1.Date;
dm.t32.Parameters.ParamValues['EndDate']:=DateTimePicker2.Date;
dm.t32.Open;
end;
procedure
TForm38.ToolButton4Click(Sender: TObject);
begin
dm.t32.Close;
dm.t32.Open;
end;
procedure
TForm38.ToolButton8Click(Sender: TObject);
begin
Form38.Close;
end;
procedure
TForm38.ToolButton7Click(Sender: TObject);
begin
Form34.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName) + 'Помощь.БШ');
Form34.show;
end;
end.
unit Unit39;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, cxGraphics, cxControls, cxLookAndFeels, cxLookAndFeelPainters,
cxStyles, dxSkinsCore, dxSkinscxPCPainter, cxCustomData, cxFilter,
cxData, cxDataStorage, cxEdit, DB, cxDBData, cxGridCustomTableview,
cxGridTableView, cxGridDBTableView, ImgList, ComCtrls, ToolWin, RpCon,
RpConDS, RpDefine, RpRave, StdCtrls, ExtCtrls, cxGridLevel, cxClasses,
cxGridCustomView, cxGrid, cxDBLookupComboBox;
type
TForm39 = class(TForm)
cxGrid2: TcxGrid;
cxGridDBTableView1:
TcxGridDBTableView;
cxGridLevel1: TcxGridLevel;
Panel2: TPanel;
Label2: TLabel;
Label3: TLabel;
DateTimePicker1:
TDateTimePicker;
DateTimePicker2:
TDateTimePicker;
Button2: TButton;
RvProject30: TRvProject;
RvDataSetConnection301:
TRvDataSetConnection;
RvDataSetConnection302:
TRvDataSetConnection;

```

```

RvDataSetConnection303:
TRvDataSetConnection;
RvDataSetConnection304: TRvDataSetConnection;
ToolBar1: TToolBar;
ToolButton4: TToolButton;
ToolButton5: TToolButton;
ToolButton7: TToolButton;
ToolButton8: TToolButton;
ImageList1: TImageList;
cxGridDBTableView1id_prod: TcxGridDBCColumn;
cxGridDBTableView1sumppkolichestvo: TcxGridDBCColumn;
cxGridDBTableView1sumppsumma: TcxGridDBCColumn;
cxGridDBTableView1id_kateg: TcxGridDBCColumn;
cxGridDBTableView1id_sklada: TcxGridDBCColumn;
cxGridDBTableView1data: TcxGridDBCColumn;
ToolButton1: TToolButton;
cxGridDBTableView1Column1: TcxGridDBCColumn;
procedure Button2Click(Sender: TObject);
procedure ToolButton4Click(Sender: TObject);
procedure ToolButton7Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations } end;
var
Form39: TForm39;
implementation
uses Unit5, Unit34;
{$R *.dfm}
procedure
TForm39.Button2Click(Sender: TObject);
var StartDate, EndDate: string;
begin
StartDate := formatDateTime('yyy y-mm-dd',DateTimePicker1.Date);
EndDate := formatDateTime('yyy y-mm-dd',DateTimePicker2.Date);
dm.t33.Close;
dm.t33.SQL.Clear;
dm.t33.SQL.Text := 'SELECT * FROM zapros6 z where (z.data between :StartDate AND :EndDate)';
dm.t33.Parameters.ParamByName('StartDate').DataType := ftDate;
dm.t33.Parameters.ParamByName('EndDate').DataType := ftDate;
dm.t33.Parameters.ParamValues['StartDate'] := DateTimePicker1.Date;
dm.t33.Parameters.ParamValues['EndDate'] := DateTimePicker2.Date;
dm.t33.Open;
end;
procedure
TForm39.ToolButton4Click(Sender: TObject);
begin
dm.t33.Close;
dm.t33.Open;
end;
procedure
TForm39.ToolButton7Click(Sender: TObject);
begin
Form34.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName) + 'Помощь.Бит');
Form34.show;
end;
end.

```

```

unit Unit40;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, cxGraphics, cxCon- trols, cxLookAndFeels, cxLookAndFeelPainters,
cxStyles, dxSkinsCore, dxSkinscxPCPainter, cxCustomData, cxFilter,
cxData, cxDataStorage, cxEdit, DB, cxDBData, cxDBLookupCom- boBox,
cxGridCustomT ableView, cxGridTableView, cxGridDBTa- bleView, cxGridLevel,
cxClasses, cxGridCustomView, cxGrid, ImgList, ComCtrls, ToolWin;
type
TForm40 = class(TForm)
ToolBar1: TToolBar;
ToolButton4: TToolButton;
ToolButton5: TToolButton;
ToolButton6: TToolButton;
ToolButton7: TToolButton;
ToolButton8: TToolButton;
ImageList1: TImageList;
cxGrid2: TcxGrid;
cxGrid2DBTableView1:
TcxGridDBTableView;
cxGrid2Level1:
TcxGridLevel;
cxGrid2DBTableView1id_yach_hran: TcxGridDBColumn;
{$R *.dfm}
cxGrid2DBT ableView1mest_vsego: TcxGridDBColumn;
cxGrid2DBT ableView 1 zanyato:
TcxGridDBColumn;
cxGrid2DBT ableView1 svobodno:
TcxGridDBColumn;
procedure ToolBut-
ton4Click(Sender: TObject);
procedure ToolBut-
ton8Click(Sender: TObject);
procedure ToolBut-
ton7Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations } end;
var
Form40: TForm40;
implementation
uses Unit5, Unit34;
procedure
TForm40.ToolButton4Click(Sender: TObject);
begin
dm.t35.Close;
dm.t35.Open;
end;
procedure
TForm40.ToolButton8Click(Sender: TObject);
begin
Form40.Close;
end;
procedure
TForm40.ToolButton7Click(Sender: TObject);
begin
Form34.WebBrowser1 .Navigate( ExtractFilePath(Application.ExeName) + 'Помощь.htm');
Form34.show;
end;
end.

```

