

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Кибернетики

Направление подготовки 09.04.01 Информатика и вычислительная техника

Кафедра Оптимизации систем управления

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы	
Исследование современных приложений ИТ-поддержки. Разработка базы данных предварительной диспетчеризации обращений пользователей ООО «ИТ-СЕРВИС»	

УДК 004.658.6

Студент

Группа	ФИО	Подпись	Дата
8ВМ4В	Быков Александр Вадимович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент каф. ОСУ	Фофанов О.Б.	к.т.н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент каф. МЕН	Конотопский В.Ю.	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Акулов П.А.			

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ОСУ	Иванов М.А.	к.т.н.		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Кибернетики

Направление подготовки 09.04.01 Информатика и вычислительная техника

Кафедра Оптимизации систем управления

УТВЕРЖДАЮ:

Зав. кафедрой

(Подпись) _____ (Дата) Иванов М.А.
(Ф.И.О.)

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

В форме:

Магистерская диссертация

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8ВМ4В	Быков Александр Вадимович

Тема работы:

Исследование современных приложений ИТ-поддержки. Разработка базы данных предварительной диспетчеризации обращений пользователей ООО «ИТ-СЕРВИС»	
Утверждена приказом директора (дата, номер)	19.02.2016 №1321/с

Срок сдачи студентом выполненной работы:	06.06.2016
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе (наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).	Объектом исследования и проектирования является система предварительной диспетчеризации обращений от пользователей.
---	---

Перечень подлежащих исследованию, проектированию и разработке вопросов <i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i>	1. Проанализировать в современных условиях предметную область и указать на проблемы в деятельности службы; 2. Определить основные требования в необходимости автоматизации к проектируемой системе и на основе анализа предметной области сформулировать перечень задач, подлежащих к автоматизации; 3. Разработать способы корректировки норм времени согласно SLA между предприятиями.
Перечень графического материала <i>(с точным указанием обязательных чертежей)</i>	Блок-схемы работы алгоритмов, результаты работы программы.
Консультанты по разделам выпускной квалификационной работы <i>(с указанием разделов)</i>	
Раздел	Консультант
Обзор литературы Описание предметной области Концептуальная модель данных Реализация системы и описание приложения	Фофанов О.Б.
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Конотопский В.Ю.
Социальная ответственность	Акулов П.А.
Приложение А	Чердынцев Е.С.
Названия разделов, которые должны быть написаны на русском и иностранном языках:	
Раздел 4: Реализация системы	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	15.06.2015
--	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент каф. ОСУ	Фофанов О.Б.	К.Т.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ4В	Быков Александр Вадимович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8BM4B	Быков Александр Вадимович

Институт	Кибернетики	Кафедра	ОСУ
Уровень образования	Магистр	Направление/специальность	09.04.01 Информатика и вычислительная техника

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого и инновационного потенциала НТИ	
2. Разработка устава научно-технического проекта	
3. Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	
4. Определение ресурсной, финансовой, экономической эффективности	

Перечень графического материала (с точным указанием обязательных чертежей):

1. «Портрет» потребителя результатов НТИ
2. Сегментирование рынка
3. Оценка конкурентоспособности технических решений
4. Диаграмма FAST
5. Матрица SWOT
6. График проведения и бюджет НТИ
7. Оценка ресурсной, финансовой и экономической эффективности НТИ
8. Потенциальные риски

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Конотопский Владимир Юрьевич	к.э.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8BM4B	Быков Александр Вадимович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ВМ4В	Быков Александр Вадимович

Институт	Кибернетики	Кафедра	ОСУ
Уровень образования	Магистр	Направление/специальность	09.04.01 Информатика и вычислительная техника

Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	<i>В данной работе проводились исследования и создание автоматизированной ИС для диспетчеризации обращений пользователей, а именно создания базы данных для пользователей. Область применения – информационные технологии.</i>
--	--

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Производственная безопасность 1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности. 1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности. 1.3. Рекомендации по минимизации влияния	<i>1.1. В качестве вредных факторов выделены: шум и электромагнитное излучение. 1.2. В качестве опасных факторов выделены: возможность поражения электрическим током и возникновение пожара, электромагнитного излучения. 1.3. Приведены рекомендации по улучшению микроклимата в офисном помещении, а также рекомендации по минимизации влияния шума, электромагнитного излучения и освещения, меры по обеспечению пожарной безопасности, способы защиты от электрического тока.</i>
2. Экологическая безопасность: 2.1. Анализ воздействия на окружающую среду 2.2. Рекомендации по минимизации влияния на окружающую среду	<i>2.1. Деятельность организации не связана с производством, следовательно, влияние на окружающую среду минимально. 2.2. Рассмотрена утилизация бумажных отходов и неисправных комплектующих ПК.</i>
3. Безопасность в чрезвычайных ситуациях: 3.1. Перечень возможных ЧС на объекте 3.2. Меры по ликвидации ЧС и последствий	<i>3.1. Основные ЧС в офисном помещении является возникновение пожара, также при хранении конфиденциальных данных в электронных таблицах можно говорить о возможности возникновения кибертерроризма. 3.2. Приведены способы защиты от пожара и кибератак.</i>
4. Правовые и организационные вопросы обеспечения безопасности: 4.1. Психофизиологические факторы. 4.2. Организационные мероприятия при компоновке рабочей зоны. 4.3. Обеспечение гарантий защиты конфиденциальных данных граждан	<i>4.1. Рассмотрены психофизиологические факторы, 4.2. Организационные мероприятия при компоновке рабочей 4.3. Обеспечение гарантий защиты конфиденциальных данных граждан с помощью комплекса технических и юридических мер.</i>

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. экологии и безопасности жизнедеятельности	Акулов Петр Анатольевич			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ4В	Быков Александр Вадимович		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Кибернетики

Направление подготовки (специальность) 09.04.01 Информатика и вычислительная техника

Уровень образования – бакалавр

Кафедра Оптимизации система управления

Период выполнения – осенний / весенний семестр 2015/2016 учебного года

Форма представления работы:

Магистерская диссертация

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:

06.06.2016

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
15.06.2015	<i>Обзор литературы</i>	10
29.06.2016	<i>Описание предметной области и выбор концептуальной модели данных</i>	15
20.09.2015	<i>Обоснование выбора сервера базы данных и инструментальной среды разработки</i>	10
15.04.2016	<i>Реализация системы</i>	25
25.05.2016	<i>Описание приложения</i>	10
21.05.2016	<i>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</i>	15
06.05.2015	<i>Социальная ответственность</i>	10
30.05.2015	<i>Приложение А. Circuit Implementation Procedure</i>	5

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент каф. ОСУ	Фофанов О.Б.	к.т.н.		

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
зав. каф. ОСУ	Иванов М.А.	к.т.н.		

ЗАПЛАНИРОВАННЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ПРОГРАММЕ

Код Результата	Результат обучения (выпускник должен быть готов)
<i>Профессиональные компетенции</i>	
P1	Применять глубокие естественнонаучные и математические знания для решения научных и инженерных задач в области информатики и вычислительной техники.
P2	Применять глубокие специальные знания в области информатики и вычислительной техники для решения междисциплинарных инженерных задач.
P3	Ставить и решать инновационные задачи инженерного анализа, связанные с созданием аппаратных и программных средств информационных и автоматизированных систем, с использованием аналитических методов и сложных моделей.
P4	Выполнять инновационные инженерные проекты по разработке аппаратных и программных средств автоматизированных систем различного назначения с использованием современных методов проектирования, систем автоматизированного проектирования, передового опыта разработки конкурентно способных изделий.
P5	Планировать и проводить теоретические и экспериментальные исследования в области проектирования аппаратных и программных средств автоматизированных систем с использованием новейших достижений науки и техники, передового отечественного и зарубежного опыта. Критически оценивать полученные данные и делать выводы.
P6	Осуществлять авторское сопровождение процессов проектирования, внедрения и эксплуатации аппаратных и программных средств автоматизированных систем различного назначения.
<i>Универсальные компетенции</i>	
P7	Использовать глубокие знания по проектному менеджменту для ведения инновационной инженерной деятельности с учетом юридических аспектов защиты интеллектуальной собственности.
P8	Осуществлять коммуникации в профессиональной среде и в обществе в целом, активно владеть иностранным языком, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности, в том числе на иностранном языке.
P9	Эффективно работать индивидуально и в качестве члена и руководителя группы, в том числе междисциплинарной и международной, при решении инновационных инженерных задач.
P10	Демонстрировать личную ответственность и ответственность за работу возглавляемого коллектива, приверженность и готовность следовать профессиональной этике и нормам ведения инновационной инженерной деятельности. Демонстрировать глубокие знания правовых, социальных, экологических и культурных аспектов инновационной инженерной деятельности.
P11	Демонстрировать способность к самостоятельному обучению, непрерывному самосовершенствованию в инженерной деятельности, способность к педагогической деятельности.

РЕФЕРАТ

Выпускная диссертационная работа содержит 136 с., 15 рис., 17 табл., 31 источника, 7 прил.

Ключевые слова: база данных, СУБД, Oracle, настольные СУБД, серверные СУБД, информационная система.

Цель работы – провести теоретические и практические исследования современных приложений ИТ-поддержки и разработать базу данных предварительной диспетчеризации обращений пользователей ООО «ИТ-СЕРВИС».

Объектом исследования и проектирования является система предварительной диспетчеризации обращений от пользователей, формирование SLA на техническое обслуживание пользователей различного типа.

Практическая значимость исследования заключается в том, что полученные результаты в диссертационной работе и предоставленная версия системы будут применены в работе специалистами технической поддержки для обслуживания пользователей.

Выпускная квалификационная работа выполнена в соответствии с требованиями стандарта ТПУ (СТО ТПУ 2.5.01-2014) в текстовом редакторе Microsoft Word 2010.

Перечень условных обозначений, единиц и терминов

Авторизация - процесс проверки прав доступа при попытке входа в систему.

Аутентификация - проверка подлинности предъявленного пользователем идентификатора.

База данных - программное обеспечение, т.е. информационная модель необходимая для хранения, обработки и получения информации о группе объектов.

Система управления базами данных - совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Full Time Equivalent – это эквивалент полной занятости, т.е. это единица измерения рабочего времени.

Service Level Agreement – это соглашение об уровне поставленной услуге, т.е. формальный договор между заказчиком услуги и её поставщиком, который в себе содержит описание услуги, права, а также самое главное, согласованный уровень качества предоставления необходимой услуги.

АРМ – автоматизированное рабочее место.

БД – база данных.

ИС – информационная система.

ПК – персональный компьютер.

ПО – программное обеспечение.

СУБД – система управления базами данных.

FTE - Full Time Equivalent.

SLA – Service Level Agreement.

Содержание

Введение.....	13
1.Обзор	литературы
.....	16
1.1.Настольные СУБД	16
1.2.Серверные СУБД	17
2.Описание предметной области и выбор концептуальной модели данных	20
2.1.Проведение анализа предметной области	20
2.2. Выбор концептуальной модели данных	24
2.3. IDEF1х модель базы данных.....	28
3.Обоснование выбора сервера базы данных и инструментальной среды разработки.....	29
3.1.Обоснование выбора СУБД.....	29
3.2.Обзор основных характеристик разных СУБД.....	30
3.3.Сравнительные характеристики СУБД	35
3.4.Обоснование выбора инструментальной среды разработки	36
3.5.Сравнение и выбор средства разработки.....	39
4.Реализация системы.....	41
4.1.Выбор типа сетевой архитектуры	41
4.2.Описание ограничений целостности БД	42
4.3. Администрирование баз данных	44
4.3.1.Защита от несанкционированного доступа	44
4.3.2.Администрирование пользователей и организация прав доступа клиентов к данным Oracle	48
4.4.Требования к составу и параметрам технических средств.....	52
5.Описание приложения.....	53
5.1.Описание приложения администрирования пользователей.....	53
5.2.Описание клиентской части приложения.....	55

6.Финансовый менеджмент, ресурсо-эффективность и ресурсосбережение	58
6.1.Выбор аналога сравнения и необходимость замены на разрабатываемый продукт.....	59
6.2.Организация и планирование работы	59
6.2.1.Продолжительность этапов работ	60
6.2.2.Расчет накопления готовности проекта.....	64
6.3.Расчет сметы затрат на выполнение проекта	65
6.3.1.Расчет заработной платы.....	66
6.3.2.Расчет затрат на социальный налог	67
6.3.3.Расчет затрат на электроэнергию	67
6.3.4.Расчет амортизационных расходов	68
6.3.5.Расчет прочих расходов	70
6.3.6.Расчет общей себестоимости разработки.....	70
6.3.7.Расчет прибыли	71
6.3.8.Расчет НДС	71
6.3.9.Цена разработки НИР	71
6.4.Оценка экономической эффективности проекта.....	71
6.5.Оценка научно-технического уровня НИР.....	72
7.Социальная	75
7.1. Производственная безопасность	76
7.1.1. Описание рабочего места.....	76
7.1.2. Микроклимат.....	76
7.1.3. Шум	77
7.1.4. Электромагнитные излучения	78
7.1.5.Освещение	79
7.2.Анализ опасных факторов.....	80
7.2.1. Пожарная безопасность.....	81
7.2.2. Электробезопасность.....	81

7.3.Рекомендации по минимизации влияния	82
7.3.1.Рекомендации по улучшению микроклимата	82
7.3.2.Рекомендации по минимизации влияния шума.....	83
7.3.3.Рекомендации по минимизации влияния электромагнитных излучений.....	83
7.3.4. Рекомендации по минимизации влияния освещения.....	83
7.3.5.Меры по обеспечению пожарной безопасности.....	83
7.3.6. Рекомендации по защите от электрического тока.....	84
7.4 Экологическая безопасность	86
7.4.1.Анализ воздействия на окружающую среду	86
7.4.2.Рекомендации по минимизации влияния на окружающую среду ..	86
7.5. Безопасность в чрезвычайных ситуациях	87
7.5.1.Перечень возможных чрезвычайных ситуаций на объекте.....	87
7.5.2.Меры по ликвидации чрезвычайных ситуаций и последствий.....	88
7.6.Правовые и организационные вопросы обеспечения безопасности .	90
7.6.1.Психофизиологические факторы	90
7.6.2.Организационные мероприятия при компоновке рабочей зоны	90
7.6.3.Обеспечение гарантий защиты конфиденциальных данных граждан	91
Заключение	93
Список публикаций.....	95
Список использованных источников	96
Приложение А	99
Приложение Б.....	110
Приложение В	112
Приложение Г	114
Приложение Д	116
Приложение Е.....	117
Приложение Ж	120

Введение

В наше время происходит стремительное развитие информационных технологий. На протяжении последних лет производится переход от кропотливого ручного труда к автоматизированному или частично автоматизированному труду.

На предприятиях, организациях повсеместно используются персональные ЭВМ, поэтому многие организации и частные лица используют электронные базы данных для поддержки своих рабочих процессов. В основном раньше это были системы на одного - двух пользователей. Обычно такой ряд БД используется независимо друг от друга, если данная, хранимая информация в таких БД предоставляет заинтересованность не только для непосредственных пользователей, то для дальнейшего её распространения использовались бумажные отчеты или справки, которые были созданы в БД.

Большие объёмы информации невозможно проработать без специальных средств машинной обработки. В последнее время широкое распространение получили автоматизированные информационные системы: информационно-справочные, информационно-поисковые. Слабая система учета порождает массу ошибок обработки обращений. Поэтому данная проблема влечет за собой создание задержек выполнения работ по устранению причин обращений, т.е. если сказать другими словами, то потенциальную возможность утери информации, невозможность своевременного поиска и фильтрации данных и, как следствие, дополнительные расходы человеческих и денежных ресурсов.

В настоящее время все автоматизированные информационные системы предназначены для регистрации, хранения и обработки данных с целью поиска и выдачи ответов на запросы пользователей.

БД и интерфейсы к ним становятся наиболее актуальной и динамически развивающейся областью в сфере информационных технологий, что в принципе и неудивительно. В связи с быстрым развитием общества требуется большое количество хранимых структурированных данных во всех областях современной жизни. База данных - это хранилище структурированных данных,

причем данные необходимо, чтобы были непротиворечивыми, минимально избыточными и целостными.

База данных создается для необходимости хранения данных, которые содержат в себе необходимые сведения об изучаемой предметной области, т.е. некоторой области человеческой деятельности или области реального мира. База данных, которая относится к одной и той же предметной области, в различных случаях может содержать в себе разную детализированную информацию о предметной области. Потому что степень детализации определяется множеством факторов, прежде всего это цель использования информации из БД и сложностью функционально производственных (деловых) процессов, существующих в пределах рассматриваемой предметной области в конкретных условиях.

Цель работы – провести теоретические и практические исследования современных приложений ИТ-поддержки и разработать базу данных предварительной диспетчеризации обращений пользователей ООО «ИТ-СЕРВИС».

Для достижения данной цели необходимо решить следующие задачи:

1. Проанализировать в современных условиях предметную область и указать на проблемы в деятельности работников службы;
2. Определить основные требования в необходимости автоматизации к проектируемой системе и на основе анализа предметной области сформулировать перечень задач, подлежащих к автоматизации;
3. Разработать способы корректировки норм времени, согласно SLA между предприятиями;

Объектом исследования и проектирования является система предварительной диспетчеризации обращений от пользователей, формирование корректировки норм времени согласно SLA между предприятиями на техническое обслуживание пользователей различного типа.

Предметом исследования в данной работе является область исследования, которая является объектом некоторой деятельности, а именно деятельность компании ООО «ИТ-СЕРВИС», отдел «Единая диспетчерская корпоративная служба» (далее именуемый ЕКДС).

Научная новизна диссертационной работы заключается в следующем:

1. Определены критерии выбора способа настройки функциональных интерфейсов в зависимости от сложности реализуемых преобразований;
2. Произведены усовершенствование модели сущность-связь, которая отличается от используемой ранее в компании, тем, что данная модель позволяет однозначно идентифицировать сущности и атрибуты сущностей на основе необходимых функциональных требований;
3. Реализация данного проекта позволит организовать «Единую точку входа», которая в дальнейшем отразится на сокращении ручного труда специалистов 1 линии компании ООО «ИТ-СЕРВИС» и позволит собрать полную статистику трудозатрат всех смежных функциональных групп для расчета показателя FTE.

Теоретическая значимость исследования заключается в том, что полученные результаты в данной работе вносят вклад в развитие и создание новой автоматизированной информационной системы для сокращения ручного труда сотрудников компании ООО «ИТ-СЕРВИС».

Практическая значимость исследования заключается в том, что полученные в диссертационной работе результаты и предоставленная версия системы будут применены в работе специалистами технической поддержки для обслуживания пользователей предприятий ПАО «СИБУР Холдинг».

1. Обзор литературы

В данном разделе проводится аналитика, рассматриваются наиболее актуальные на текущий момент системы управления базами данных, также в данном разделе дается определение термину и приводится краткий обзор характеристики СУБД и выделяются основные группы СУБД.

Анализ литературы показал, что наиболее часто употребляемое понятие определения «система управления базами данных (СУБД)» является рассмотрение СУБД, как совокупность программных и лингвистических средств общего или специального назначения, которые могут обеспечивать управление, создание и использование баз данных.

После проведения аналитики по обзору литературы были выявлены значимые функции СУБД:

1. Управление данными во внешней памяти (на дисках);
2. Управление данными в оперативной памяти с использованием дискового кэша;
3. Журнализация изменений, резервное копирование и восстановление базы данных после сбоев;
4. Поддержка языков БД.

На текущий момент выделяют 2 основные группы СУБД:

1. Настольные;
2. Серверные;

Далее более подробно будут рассмотрены основные группы СУБД.

1.1. Настольные СУБД

Настольные СУБД необходимы для использования относительно небольших задач, т.е. объем обрабатываемых данных должен быть небольшим, количество пользователей также должно быть небольшим. Исходя из всего выше сказанного, данные СУБД обладают сравнительно упрощенной архитектурой, поэтому СУБД работают в режиме файл-сервер. Основным недостатком заключается в том, что не поддерживаются исключительно

некоторые необходимые функции. Например, не ведется журнал транзакций, отсутствует автоматическое восстановление базы данных после сбоя.

Стоит отметить еще тот факт, что такие системы имеют достаточно большую сферу деятельности (например, государственные [муниципальные] учреждения, сфера образования, обслуживания, средний и малый бизнес). Основанная специфика возникающих задач в данных сферах заключается в том, что объемы таких данных малые, частота обновлений невысокая, предприятие обычно расположено в одном небольшом здании, следовательно, из-за этого и количество пользователей, тоже небольшое, примерно от 1 до 15 человек. Поэтому для таких подобных условий использование настольных СУБД для управления информационными системами является вполне оправданным решением.

Одними из первых СУБД были dBase-совместимые программные системы, разработанные разными фирмами. Первой широко распространенной системой такого рода была система dBase III – PLUS [2].

На текущий момент широкое распространение получила СУБД Microsoft Access, данная СУБД входит в основной пакет Microsoft Office.

1.2.Серверные СУБД

В отличие от настольных СУБД, серверные СУБД необходимы для крупных организаций, потому что использование файл-серверных технологий уже считаются неудовлетворительным. Поэтому на первых ролях борьбы за автоматизацию систем находятся серверные СУБД.

Крупнейшими производителями таких систем предназначенных для обработки и хранения данных являются 3 корпорации: Oracle, Microsoft и IBM. Диаграмма соотношения объемов продаж соответствующих систем приводится на рисунке 1.

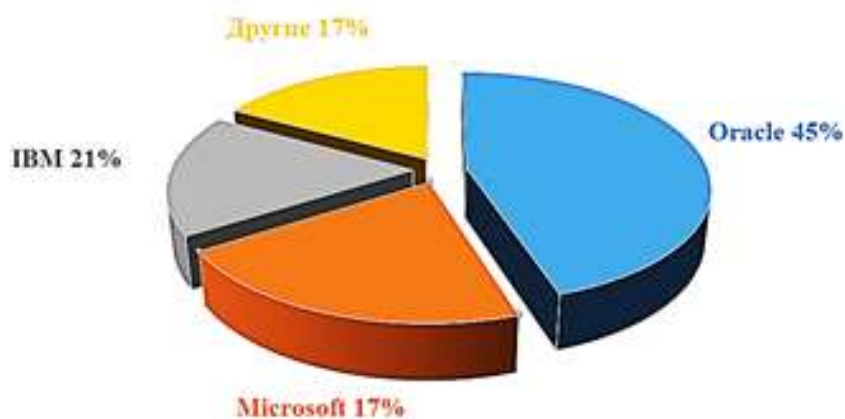


Рисунок 1 – Диаграмма отношения объемов продаж

Самыми распространенными «клиент-серверными» системами являются системы Oracle (разработчик компания Oracle), MS SQL Server (разработчик компания Microsoft), DB2, Informix Dynamic Server (компания IBM) [1].

На текущий момент разработано несколько версий систем Oracle, каждая из которых включает, целую линейку продуктов, например Oracle 8, Oracle 9i, Oracle 10g.

Естественно, что соответствующие линейки продуктов Oracle, также и включают в себя, как собственно СУБД (пример Oracle Database 10g, Oracle Database 11g), так и средства разработки и анализа данных.

Компания Oracle предлагает комплексные, открытые, доступные и удобные в использовании технологические решения. Готовые пакетируемые решения автоматически включают в свою стоимость базу данных, сервер приложений, интеграционную платформу, инструменты аналитики и управления неструктурированными данными. Масштабируемые бизнес-приложения Oracle могут быть легко интегрированы с ИТ-инфраструктурой предприятия без потери уже вложенных в ИТ инвестиций [2].

СУБД Oracle Database 11g обеспечивает улучшенные характеристики за счет автоматизации задач администрирования и обеспечения, лучших в отрасли возможностей по безопасности и соответствию нормативно-правовым актам в области защиты информации. Появилось больше функций автоматизации, самодиагностики и управления. Среди характеристик системы можно отметить

управление большими объемами данных с использованием распределенных таблиц и компрессии, эффективную защиту данных, возможность полного восстановления, возможность интеграции геофизических данных медиа-контента в бизнес-процесс [2].

Универсальный сервер БД DB2 Universal Database – это объектно-реляционная система управления базами данных с поддержкой мультимедиа и Web, работающая на системах от ПК и серверов на процессорах Intel до Unix, от простейших однопроцессорных систем до симметричных многопроцессорных систем. DB2 Universal Database объединяет в себе высокую производительность систем обработки операций в режиме on-line, модернизированные средства оптимизации с возможностями параллельной обработки баз данных, объектно-реляционные расширения [1].

Подводя итог обзора, стоит отметить, что современные СУБД не удовлетворяют полному объему требований пользователей. Я считаю, что нельзя, да и невозможно использовать одну СУБД во всех сферах деятельности. Потому что некоторая часть СУБД имеют свою предрасположенность к определенной задаче, как пример привести - это MS Access в образовательном процессе. Данная СУБД прекрасно позволяет научиться или обучиться создавать простейшие базы данных для отработки навыков и в дальнейшем позволит использовать весь наработанный опыт для создания более сложных СУБД на предприятиях в различных отраслях промышленности и в разной сфере услуг [2].

2. Описание предметной области и выбор концептуальной модели данных

2.1. Проведение анализа предметной области

Предметная область — множество всех предметов, свойства которых и отношения, между которыми рассматриваются в научной теории.

Предметная область — часть реального мира, рассматриваемая в пределах данного контекста. Под контекстом может пониматься, например, область исследования или область, которая является объектом некоторой деятельности.

Рассмотрим предметную область - компания ООО «ИТ-СЕРВИС» отдел «Единая диспетчерская корпоративная служба».

После детального изучения предметной были выявлены основные аспекты данной предметной области. В обязанности специалиста отдела ЕКДС входит следующие действия:

1. Регистрирование в системе обращений/нарядов/изменений/проблем от пользователей, у которых возникают различные проблемы в сфере ИТ-услуг и связи;
2. Проведение консультации на 1 линии телефонным звонком, тем самым повышение личного показателя First Call Request (FLC) и First Line Resolution (FLR);
3. Привлечение специалистов различных сфер деятельности для детального и быстрого закрытия обращений/нарядов/ изменений/проблем;
4. Проведение мониторинга на доступность информационных систем относящихся к «критичным» системам для сервиса, в случае недоступности регистрирование «критичного» инцидента в системе;
5. Изменять статус в зависимости от этапа, на котором находится данное обращение, а именно «в работе/ожидание/выполнено/решение предоставлено/закрыто».

Все выше перечисленные действия сотрудник ЕКДС должен отображать в автоматизированной системе для дальнейшего учета трудозатрат и расчета показателя FTE.

Поэтому с каждой организацией ПАО «СИБУР-Холдинг» заключается договор (SLA), согласно которому, поставляются услуги для определенной организации, которая в дальнейшем позиционирует себя, как заказчик. В данном договоре подробно приведено описание услуги, права, а также самый важный аспект данного договора, согласованный уровень качества предоставления необходимой услуги, согласно которому заказчик вправе требовать со специалистов ИТ-поддержки пояснения в случае неудовлетворительного качества выполнения работ по услуге.

В рамках работы технической поддержки выполняются обращения/наряды/изменения/проблемы, которые связаны с вопросами в сфере ИТ-услуг и связи и определены регламентом, а именно согласно СТП СР 95-П01 (Порядок обработки обращений при предоставлении услуге в области информационных технологий) от 03.07.2012 г. редакция 1.0.

Работа с пользователями (инициаторами) начинается с момента подачи обращения по электронной почте или телефонным звонком с той или иной проблемой, которая действительно связана с проблемой относящиеся к сфере ИТ-услуг и связи. Поэтому при заведении в системе обращений/ нарядов/ изменений/ проблем учитывается множество различных факторов:

1. Предприятие инициатора;
2. Номер договора, согласно которому оказываются услуги в сфере ИТ-услуги и связи;
3. Тип пользователя;
4. Приоритет;
5. Классификации.

Рассмотрим более подробно, как рассчитывается крайний срок, и из-за каких главных аспектов SLA крайний срок зависим.

При заведении в системе обращений/нарядов/изменений/проблем согласно выбранной услуге проставляется автоматически системой крайний срок, согласно регламенту - установленным уровням обслуживания (SLA) для каждой организации крайний срок на решение вопроса или проблемы относящихся к сфере ИТ-услуг и связи он свой.

На выставление значения крайний срок, влияют следующие факторы:

1. Тип пользователя (ВИП-пользователь, Ключевой пользователь, Обычный пользователь);
2. Приоритет (низкий, средний, высокий).
3. Классификация (запрос на обслуживание/ запрос на изменение/ консультация/ жалоба);

Данные факторы напрямую влияют на крайний срок. Во-первых, тип пользователя, важен тем, что проблемы могут возникать не только у обычных сотрудников предприятия, но и у директора предприятия, руководителей направлений и если не решить данные проблемы в кратчайшие сроки могут возникнуть проблемы, которые могут повлечь за собой крупные убытки для бизнеса. Во-вторых, приоритет и классификация, важны тем, что данные факторы напрямую влияют на расчет крайнего срока. Согласно договорам, заключенным между предприятиями холдинга, сказано, что на решение «запроса на обслуживание» и оказания «консультации» в рамках обращения, отводится 16 рабочих часов, на решение «среднего инцидента» (Инцидент, который носит единичный характер) отводится 8 рабочих часов, а если инцидент относится к массовому или пострадал один или несколько критичных сервисов компании, то выставляется «высокий-инцидент» и на его решение отводится 1 рабочий час. Стоит отметить, что данные метрики распространяются на предприятия, которые расположены, как правило, в западной части России, а предприятия, которые располагаются за Уралом, как правило, метрики увеличиваются, это связано с разными факторами. Такими, как количество подписанных услуг за предприятием, количество работающих сотрудников на предприятии, количество «критичных» систем внедренных в

производство на предприятии. Классификация «запрос на изменения», как правило, особо точной метрики не имеет, т.к. данный запрос относится к длительному процессу выполнения работ и выставляется из заполненного документа пользователем (данный документ изначально должен пройти 3 стадии согласования, после чего отправляется в техническую поддержку) по которому и запускается процесс «запрос на изменение». Процесс изменения, как правило, связан с внедрением нового модуля в ИС, поэтому данные работы выполняются чаще всего подрядными организациями и на них не распространяются метрики, которые заключены в SLA. Тип классификации «Жалоба». На рассмотрение претензии от пользователя организации отводится 40 рабочих часов. В течение данного времени экспертами проверяется порядок выполнения всех этапов жизненного цикла обращения, проводится полный разбор и анализ данного обращения, выявляются, неучтенные моменты при запуске данной услуги или основные ошибки исполнителя по обращению и в заключение как, правило, проводятся мероприятия для дальнейшего верного и качественного выполнения работ.

При проведении анализа предметной области, было замечено, что много ручной работы проводится сотрудниками 1 линии, а именно дублирование информации.

В рамках нового проекта требуется реализовать связывание нескольких форм (обращение-наряд/изменение/проблема). Данная реализация позволит создать «Единую точку входа», следовательно, в рамках одного обращения будет возможно получить всю необходимую информацию, а не проводить анализ по обращению или десяткам различных обращений. Также с помощью данной реализации вся необходимая информация о работе смежных функциональных групп будет заключена в рамках одного обращения, что позволит собирать полную статистику из одного источника о проделанной работе и проводить более точные расчеты показателя FTE.

2.2. Выбор концептуальной модели данных

Для выполнения дипломной работы необходимо создать модель и провести её исследование. Изначально перед созданием модели необходимо конкретизировать цели модели моделирования. После исследования необходимо выполнить обработку и анализ результатов моделирования.

IDEF1X – это метод, который необходим для разработки реляционных баз данных. IDEF1X в своей структуре использует условный синтаксис, который специально разработан для удобного построения концептуальной схемы. Концептуальной схемой называется универсальное представление структуры данных в рамках коммерческого предприятия, независимое от конечной реализации базы данных и аппаратной платформы. Метод IDEF1X изначально при создании не был предназначен для динамического анализа по принципу "AS IS", но все же, иногда он применяется в этом качестве, как альтернатива методу IDEF1.

Для построения логической структуры БД и решение о внедрении реляционной базы данных, как части корпоративной информационной системы был выбран метод IDEF1X.

В настоящее время есть несколько причин, по которым IDEF1X не следует применять в случае построения нереляционных систем. Во-первых, метод IDEF1X обязывает проектировщика (исполнителя) определить все ключевые атрибуты, для того чтобы отличить одну сущность от другой, в то время как объектно-ориентированные системы не требуют задания ключевых ключей, в целях идентифицирования объектов. Во-вторых, не стоит исключать исключения, т.к. есть случаи, когда более чем один атрибут однозначно идентифицирует сущность, но проектировщик (исполнитель) должен определить один из этих атрибутов первичным ключом, а все остальные вторичными. В итоге сделаем вывод построена проектировщиком IDEF1X-модель и передана дальше для финальной реализации программисту является некорректной для применения методов объектно-ориентированной реализации, и предназначена для построения реляционной системы.

Если подробно рассматривать терминологию метода IDEF1X, то она практически совпадает с терминологией IDEF1, но все-таки существует ряд основных отличий в теоретических концепциях. Сущность в IDEF1X описывает совокупность или так скажем набор экземпляров, которые похожи по свойствам, но все же однозначно отличаемы друг от друга по одному или нескольким признакам, т.к. каждый экземпляр является реализацией сущности. Следовательно, сущность в IDEF1X описывает конкретный набор экземпляров части реального мира, в отличие от сущности в IDEF1, которая представляет собой абстрактный набор информационных отображений реального мира. Как пример сущности IDEF1X может быть сущность «СОТРУДНИК», в которой предоставлены все сотрудники организации, а один из них, как пример Быков Александр Вадимович, является реализацией (конкретной) этой сущности. Данный пример предоставлен на рисунке 2, надо отметить, что каждый сотрудник - это экземпляр сущности «СОТРУДНИК» и содержит следующую информацию: ID сотрудника, имя сотрудника, адрес сотрудника, оклад сотрудника и т.д.

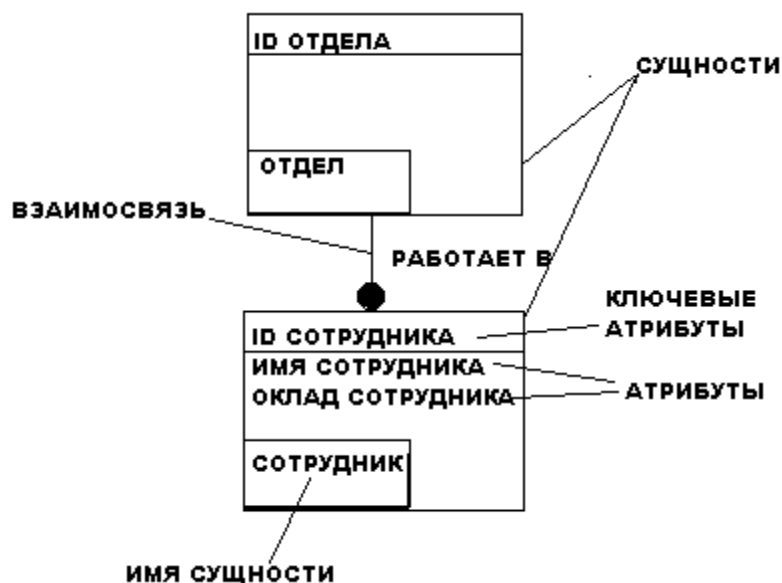


Рисунок 2 - Сущности и атрибуты в IDEF1X

В модели IDEF1X данные свойства называются атрибутами сущности. Каждый атрибут содержит в себе только часть информации о сущности.

Чтобы идентифицировать уникальную запись, для этого требуется уникальная идентификация каждой записи, в сущности, для того чтобы правильно создалась модель данных. Необходимо также запомнить, что сущности в IDEF1X всегда имеют ключевую область, следовательно, в каждой сущности необходимо, чтобы были определены ключевые атрибуты.

Очень большого внимания и концентрации требуется для выбора первичного ключа для сущности. В качестве первичных ключей, как правило, выступают несколько атрибутов или групп атрибутов. Атрибуты, которые могут быть выбраны первичными ключами, называются кандидатами в ключевые атрибуты (потенциальные атрибуты). Кандидаты в ключи должны уникально идентифицировать каждую запись сущности. В соответствии с этим, ни одна из частей ключа не может быть NULL, незаполненной или отсутствующей.

Пример, чтобы корректно использовать сущность «СОТРУДНИК» в IDEF1X модели данных (а далее в базе данных), необходимо иметь возможность уникально идентифицировать записи, как правило, по которым осуществляется выбор первичного ключа из списка предполагаемых ключей, однако могут быть применены ко всем типам баз данных и информации. Правила устанавливают, что атрибуты и группы атрибутов должны:

1. Идентифицировать уникальным образом экземпляр сущности;
2. Не использовать NULL значений;
3. Не изменяться со временем. Экземпляр идентифицируется при помощи ключа. При изменении ключа, соответственно меняется экземпляр;
4. Быть более короткими для использования индексирования и получения данных.

Для наглядного представления о том, как целесообразно выбирать первичные ключи, приведем следующий пример - выберем первичный ключ для сущности «СОТРУДНИК»:

Атрибут «ID сотрудника» является потенциальным ключом, потому что он уникален для всех экземпляров сущности «СОТРУДНИК».

Атрибут «Имя сотрудника» не подходит для потенциального ключа, так как среди работников организации может быть, к примеру, двое полностью совпадающих однофамильцев.

Атрибут «Номер страхового полиса сотрудника» является уникальным, но проблема заключается в том, что «СОТРУДНИКА» может не иметь его.

Комбинация атрибутов «Имя сотрудника» и «дата рождения сотрудника» может оказаться удачной для наших целей и стать искомым потенциальным ключом.

После проведенного анализа можно назвать два потенциальных ключа - первый «Номер сотрудника» и комбинация, включающая поля «имя сотрудника» и «Дата рождения сотрудника», но так как атрибут «Номер сотрудника» имеет самые короткие и уникальные значения, то он лучше других подходит для первичного ключа.

При выборе первичного ключа для сущности, разработчики модели часто используют дополнительный (суррогатный) ключ, т.е. произвольный номер, который уникальным образом определяет запись, в сущности. Атрибут «Номер сотрудника» является примером суррогатного ключа. Суррогатный ключ лучше всего подходит на роль первичного ключа потому, что является коротким и быстрее всего идентифицирует экземпляры в объекте. К тому же суррогатные ключи могут автоматически генерироваться системой так, чтобы нумерация была сплошной, т.е. без пропусков.

Потенциальные ключи, которые не выбраны первичными, могут быть использованы в качестве вторичных или альтернативных ключей. С помощью альтернативных ключей часто отображают различные индексы доступа к данным в конечной реализации реляционной базы.

Если сущности в IDEF1X диаграмме связаны, связь передает ключ (или набор ключевых атрибутов) дочерней сущности. Эти атрибуты называются - внешними ключами. Внешние ключи определяются как атрибуты первичных ключей родительского объекта, переданные дочернему объекту через их связь. Передаваемые атрибуты называются мигрирующими [3].

2.3. IDEF1x модель базы данных

Концептуальная модель данных построена с использованием методологии информационного моделирования IDEF1X (Integrated DEFinitions 1 eXpanded), имеющей в настоящее время статус федерального стандарта США.

Диаграмма уровня ER (Entity-Relationship) представлена на рисунке 3.

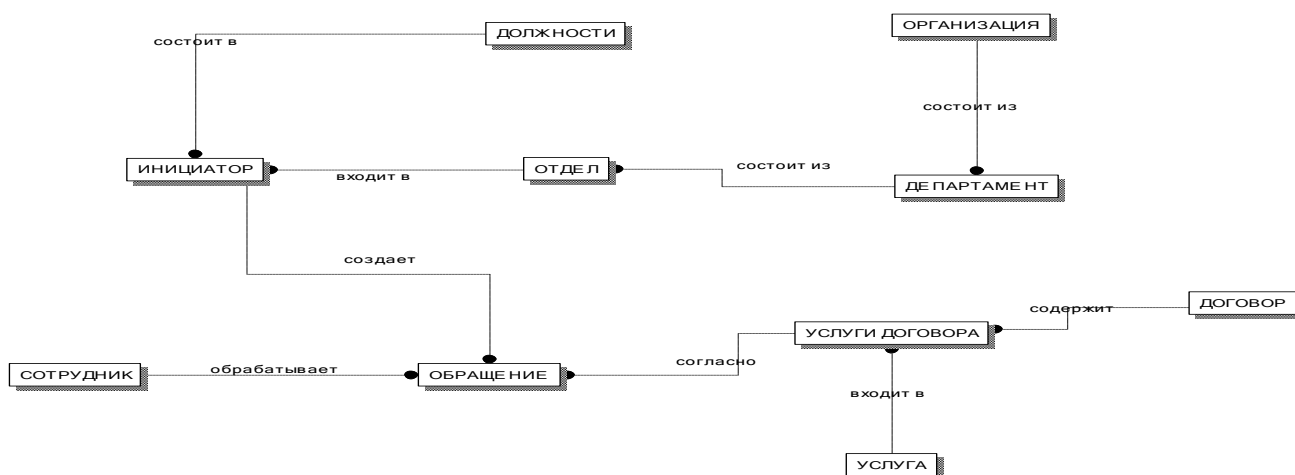


Рисунок 3 – Диаграмма ER-уровня

Диаграмма FA-уровня показывает имена всех атрибутов сущностей и связей и полностью определяет структуру и взаимосвязи данных. Включает в себя KB-уровень.

FA - диаграмму можно рассматривать, как описание реляционной структуры данных. Диаграмма уровня FA(Full-Attributed) представлена на рисунке 4.

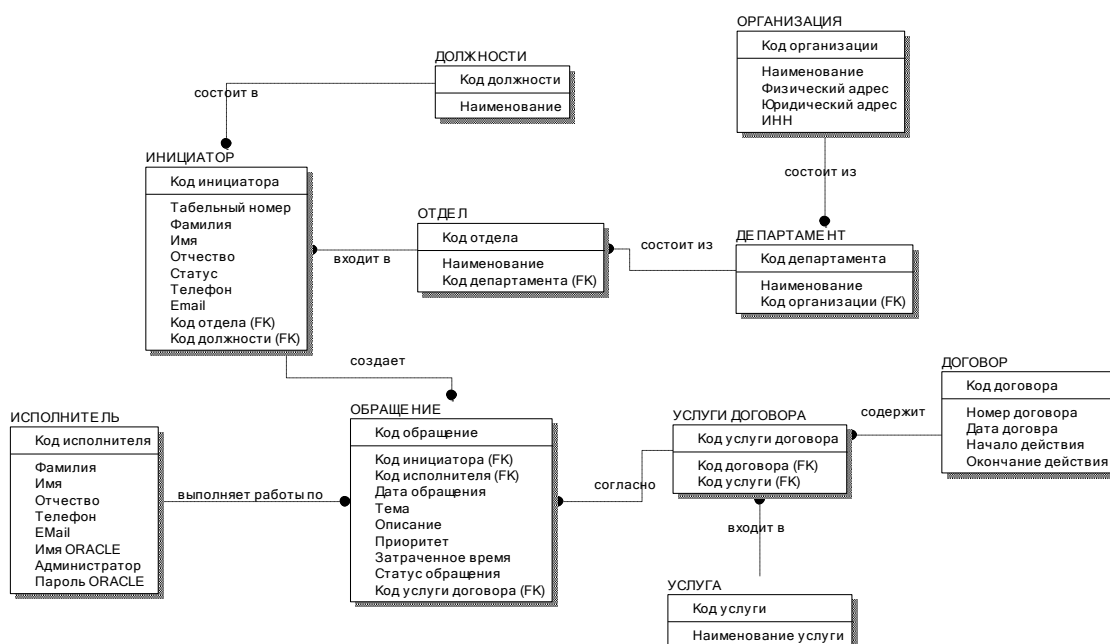


Рисунок 4 - Диаграмма FA-уровня

3. Обоснование выбора сервера базы данных и инструментальной среды разработки

3.1. Обоснование выбора СУБД

При выборе СУБД необходимо учитывать большое количество аспектов, каждый из которых накладывает определенные ограничения на выбор продукта, потому что это не только технические показатели, а также экономическая выгода и учет рынка на текущий момент.

Одной из важнейших и основной характеристики СУБД является модель данных. Если рассматривать с теоретической точки зрения любую информацию можно представить в виде реляционной модели. Необходимость реляционных баз данных заключается в том, что модель прекрасно подходит для предприятий, которые имеют немалые средства для активного внедрения новых передовых систем. Данная модель имеет проработанное математическое основание и стандарты. С точки зрения структуры данных реляционная модель данных имеет очень большую гибкость, т.к. можно изменять физическую структуру данных, не переписывая приложения.

Еще один из важных аспектов при выборе СУБД, напрямую связан с выбором аппаратных ресурсов и анализом. Стандартные тесты проводятся на современных быстродействующих системах, в то время как целью клиента может быть просто работоспособная система в рамках отведенного бюджета, также очень важны и взаимодействия с унаследованными системами, и возможности переноса накопленных данных на новую систему.

Также важным критерием выбора, если рассматривать с точки зрения перспективы становится еще и наличие эффективных средств разработки. Средства, которые обладают удобным интерфейсом, позволяют специалистам предприятия или компании самостоятельно, быстро и в соответствии с требованиями бизнеса настраивать информационные системы [4].

3.2. Обзор основных характеристик разных СУБД

Oracle Database 11g предназначена для эффективного развертывания на базе различных типов оборудования, от небольших серверов до Oracle Enterprise Grid мощных многопроцессорных серверных систем, от отдельных кластеров до корпоративных распределенных вычислительных систем [12].

Преимущества Oracle Database 11g:

1. Automatic Storage Management позволяет автоматически распределять данные между имеющимися ресурсами систем хранения данных, что повышает отказоустойчивость системы и снижает общую стоимость владения;
2. Специальные механизмы Oracle Database 11g позволяют самостоятельно перераспределять нагрузку на систему, оптимизировать и корректировать SQL-запросы, выявлять и прогнозировать ошибки;
3. Максимальный размер экземпляра базы данных Oracle может достигать 8 ЭБ;
4. Oracle Database 11g может использовать недорогие однопроцессорные компьютеры или модульные системы из «серверов-лезвий».

Firebird (FirebirdSQL) — кроссплатформенная система управления базами данных (СУБД), работающая на Mac OS X, Linux, Microsoft Windows и разнообразных Unix платформах.

Firebird используется в различных промышленных системах (складские и хозяйственные, финансовый и государственный сектора) с 2001 г. Это коммерчески независимый проект C и C++ программистов, технических советников.

Основные характеристики:

1. Соответствие требованиям ACID: Firebird сделан специально, чтобы удовлетворять требованиям «атомарности, целостности, изоляции и надёжности» транзакций («Atomicity, Consistency, Isolation and Durability»)[1];
2. Версионная архитектура. Основной особенностью Firebird — версионная архитектура, позволяющая серверу обрабатывать различные версии

одной и той же записи в любое время таким образом, что каждая транзакция видит свою версию данных, не мешая соседним («читающие транзакции не блокируют пишущие, а пишущие не блокируют читающих»). Это позволяет использовать одновременно OLTP и OLAP запросы;

3. Хранимые процедуры. Используя язык PSQL (процедурный SQL) Firebird, возможно создавать сложные хранимые процедуры для обработки данных полностью на стороне сервера. Для генерации отчётов особенно удобны хранимые процедуры с возможностью выборки, возвращающие данные в виде набора записей. Такие процедуры можно использовать в запросах точно так же как и обычные таблицы;

4. События. Хранимые процедуры и триггеры могут генерировать события, на которые может подписаться клиент. После успешного завершения транзакции (COMMIT) он будет извещён о произошедших событиях и их количестве;

5. Генераторы. Идея генераторов (последовательностей) делает возможной простую реализацию автоинкрементных полей, и не только их. Генераторы являются 64-битными хранимыми в базе данных счётчиками, работающими независимо от транзакций. Они могут быть использованы для различных целей, таких как генерация первичных ключей, управление длительными запросами в соседних транзакциях;

6. Полный контроль транзакций. Одно клиентское приложение может выполнять множество одновременных транзакций. В разных транзакциях могут быть использованы разные уровни изоляции. Протокол двухфазного подтверждения транзакций обеспечивает гарантированную устойчивость при работе с несколькими базами данных, также доступны оптимистическое блокирование данных и точки сохранения транзакций;

7. Триггеры. Для каждой таблицы возможно назначение нескольких триггеров, срабатывающих до или после вставки, обновления или удаления записей. Для триггеров используется язык PSQL, позволяя вносить начальные значения, проверять целостность данных, вызывать исключения.

В Firebird 2.5 появились «универсальные» триггеры, позволяющие в одном триггере обрабатывать вставки, обновления и удаления записей таблицы;

8. Внешние функции. Библиотеки с UDF (User Defined Function) могут быть написаны на любом языке и легко подключены к серверу в виде DLL/SO, позволяя расширять возможности сервера «изнутри»;

9. Декларативное описание ссылочной целостности. Обеспечивает непротиворечивость и целостность многоуровневых отношений «master-detail» между таблицами;

10. Набор символов. Firebird поддерживает множество международных наборов символов (включая Unicode) с множеством вариантов сортировки;

11. Соответствие стандарту SQL [5].

Microsoft Office Access — реляционная СУБД корпорации Microsoft. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Благодаря встроенному языку VBA, в самом MS Access можно писать приложения, работающие с базами данных.

Основные компоненты MS Access:

1. Построитель таблиц;
2. Построитель экранных форм;
3. Построитель SQL-запросов (язык SQL в MS Access не соответствует стандарту ANSI);
4. Построитель отчётов, выводимых на печать.

Они могут вызывать скрипты на языке VBA, поэтому MS Access позволяет разрабатывать приложения и БД практически «с нуля» или написать оболочку для внешней БД.

Microsoft Jet Database Engine, которая используется в качестве движка базы данных MS Access является файл-серверной СУБД и потому применима лишь к приложениям, работающим с небольшими объёмами данных и при небольшом числе пользователей, одновременно работающих с этими данными. Непосредственно в MS Access отсутствует ряд механизмов, необходимых в многопользовательских базах данных, таких, например, как триггеры.

MS Access, при работе с базой данных, иначе взаимодействует с жёстким (или гибким) диском, нежели другие программы.

В других программах, файл-документ, при открытии, полностью загружается в оперативную память, и новая редакция этого файла (изменённый файл) целиком записывается на диск только при нажатии кнопки «сохранить».

В MS Access новая редакция содержимого изменённой ячейки таблицы записывается на диск (сохраняется) сразу, как только курсор клавиатуры будет помещён в другую ячейку (или новая редакция изменённой записи записывается на диск сразу, как только курсор клавиатуры будет поставлен в другую запись (строку). Таким образом, если внезапно отключат электричество, то пропадёт только изменение той записи, которую не успели покинуть. Целостность данных в Access обеспечивается, также за счёт механизма транзакций.

Даже если в процессе работы с файлом базы данных не применялся режим «Конструктор» и новые данные в базу данных не добавлялись (то есть если база данных только просматривалась), то всё равно файл базы данных имеет тенденцию со временем, в процессе работы с ним, всё больше и больше увеличиваться в размере. Очень способствует увеличению размера файла применение новых сортировок и фильтров (особенно если было применено несколько разных, сильно отличающихся друг от друга сортировок/фильтров).

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется, как под GNU (General Public License), так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей, именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов

Денвер, XAMPP. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать, как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

Возможности:

1. Практически полная реализация ANSI SQL-99, плюс расширения;
2. Межплатформенная совместимость;
3. Независимые типы таблиц (MyISAM для быстрого чтения, InnoDB для транзакций и ссылочной целостности);
4. Транзакции;
5. Поддержка SSL;
6. Кэширование запросов;
7. Репликация: один головной сервер на одного подчинённого, много подчинённых на одного головного;
8. Полнотекстовая индексация и поиск с использованием типа таблиц MyISAM;
9. Внедрённая библиотека базы данных;
10. Поддержка Юникода (UTF-8);
11. Таблицы InnoDB, обеспечивающие соответствие требованиям ACID;
12. Встроенный сервер, позволяющий включать MySQL в автономные приложения;

13. Вложенные запросы и производные таблицы;
14. Новая система кодировок и сортировок;
15. Быстрый и гибкий протокол клиент-сервер с поддержкой подготовленных запросов, обеспечивающий их оптимальное исполнение;
16. Новая программа установки и настройки для Microsoft Windows и Linux;
17. Защищённые через OpenSSL соединения клиент-сервер;
18. Высоко-оптимизированная библиотека, которая может быть использована в сторонних программах;
19. Полноценная поддержка Юникода (UTF-8 и UCS2);
20. Стандартные пространственные типы данных GIS, для хранения географической информации;
21. Улучшенный полнотекстовый поиск и система помощи;
22. Хранимые процедуры и функции;
23. Обработчики ошибок;
24. Поддержка триггеров;
25. Представления;
26. Информационная схема (так называемый системный словарь, содержащий метаданные);
27. Сегментирование — возможность разбить одну большую таблицу на несколько частей, размещенных в разных файловых системах, основываясь на определенной пользователем функции. При определенных условиях это может дать серьёзное увеличение производительности и, кроме того, облегчает масштабирование таблиц.

3.3. Сравнительные характеристики СУБД

Рассмотренные платформы БД в магистерской диссертации имеют свои достоинства и недостатки. Поэтому для того, чтобы определиться с выбором, следует четко представлять себе, для каких целей она будет применяться.

Все сравнительные характеристики разных СУБД представлены в Приложении 2 в сводной таблице 1.

Оптимальным решением для выбора СУБД в моей работе является Oracle 11g. Ведь именно данная СУБД обладает огромным количеством функциональных возможностей, имеет удобные и необходимые средства администрирования удаленных серверов и, что самое важное, способна обрабатывать большие объемы данных.

3.4. Обоснование выбора инструментальной среды разработки

В настоящий день существует множество сред программирования, но все же среди них для выполнения поставленной нами задачи можно выделить следующие: Microsoft Visual Studio, C++ Builder, Microsoft Visual Basic 6.0, Codegear Delphi 2009. Рассмотрим их некоторые особенности.

Microsoft Visual Studio - универсальная среда разработки со статическими типами данных:

1. Всестороннее поддерживает множество стилей программирования (процедурное программирование, объектно-ориентированное программирование, обобщенное программирование);
2. Данная среда разработки избегает особенностей, которые зависят от платформы или не являются универсальными;
3. В данной среде не накладывается никакой избыточной нагрузки на программу, не использующую какие-либо возможности;
4. Недостаток информации о типах данных во время компиляции;
5. Язык C++ является сложным для изучения и для компиляции.

C++ Builder — система, используемая программистами для разработки программного обеспечения на языке программирования C++.

Изначально разрабатывался компанией Borland Software, а затем её подразделением CodeGear, ныне принадлежащим компании Embarcadero Technologies.

C++ Builder объединяет в себе комплекс объектных библиотек (STL, VCL, CLX, MFC и др.), компилятор, отладчик, редактор кода и многие другие компоненты.

C++ Builder содержит инструменты, которые при помощи drag-and-drop действительно делают разработку визуальной, упрощают программирование благодаря встроенному WYSIWYG — редактору интерфейса.

Delphi – это версия среды быстрой разработки приложений для Microsoft Windows. Данный продукт предназначен для более быстрой и эффективной разработки мощных Windows- и .NET-приложений баз данных.

Основные особенности:

1. Возможность полного доступа к функциям операционных систем Windows XP и Windows 7;

2. Нарастаемость за счет встраивания новых компонент и инструментов в среду Delphi;

3. Возможность разработки новых компонент и инструментов собственными средствами Delphi (существующие компоненты и инструменты доступны в исходных текстах);

4. Удобная проработка иерархии объектов;

5. Полная поддержка Unicode. Приложения могут выполняться на любой языковой версии Windows. Применение Unicode гарантирует, что приложения будут одинаково выглядеть и функционировать во всех языковых версиях Windows и поддерживать как Unicode-строки, так и ANSI-строки. Новые усовершенствованные средства локализации помогают переводить приложения на различные языки. Все функции Windows API заменены на их unicode-аналоги;

6. Новые элементы языков программирования, в том числе Generics и анонимные методы для Delphi, позволяют создавать более гибкий и качественный код и предоставляют новые возможности для рефакторинга;

7. Новая библиотека VCL включает в себя множество усовершенствований и новых компонентов для создания развитого графического интерфейса;

8. Web-библиотека VCL позволяет создавать веб-приложения с развитым интерфейсом с поддержкой AJAX;

9. Уменьшено время передачи приложением сообщений операционной системе;

10. Визуальное проектирование и разработка баз данных благодаря входящему в состав редакции Delphi Architect профессионального средства моделирования Embarcadero ER/Studio [7].

Microsoft Visual Basic 6.0 - интегрированная среда разработки программного обеспечения, разрабатываемое корпорацией Microsoft. Язык Visual Basic унаследовал дух, стиль и отчасти синтаксис своего предка — языка BASIC, у которого есть немало диалектов. В то же время Visual Basic сочетает в себе процедуры и элементы объектно-ориентированных и компонентно-ориентированных языков программирования. Интегрированная среда разработки VB включает инструменты для визуального проектирования пользовательского интерфейса, редактор кода с возможностью IntelliSense и подсветкой синтаксиса, а также инструменты для отладки приложений.

Visual Basic также является хорошим средством быстрой разработки (RAD) приложений баз данных для операционных систем семейства Microsoft Windows. Множество готовых компонентов, поставляемых вместе со средой, призваны помочь программисту сразу же начать разрабатывать бизнес-логику приложения, не отвлекая его внимание на написание кода запуска программы, подписки на события и других механизмов, которые VB реализует автоматически.

1. Поддерживает концепции объектно-ориентированного программирования;

2. Использует весь набор объектных библиотек, входящих в .NET Framework;

3. Поддерживает свободную многопоточность;

4. Неэффективный процесс разработки;

5. Скромность предоставленного разработчику ассортимента компонентов.

3.5 Сравнение и выбор средства разработки

В таблице 2 представлены основные характеристики для выбора среды разработки приложения.

Таблица 2 – Основные характеристики инструментальной среды разработки

Характеристики	Среда разработки			
	Visual Basic 6.0	Delphi	C++ Builder	MS Visual Studio
Поддерживаемые ОС	Windows	Windows	Windows	Windows
Встроенный язык	Basic	C++	Object Pascal	Visual C++,C#, Visual J#,Visual Basic .NET
Поддержка SQL	Нет	Да	Да	Да
Поддержка ООП	Да	Да	Да	Да
Наличие встроенных компонент для работы с БД Oracle	Нет	Да	Нет	Нет
Наличие сторонних компонент для работы с БД Oracle	Нет	Да	Да	Да
Встроенные средства построения отчетов	Нет	Rave Reports	Нет	Crystal Reports
Визуальные компоненты для работы с БД	Нет	Да	Да	Да

Наиболее подходящим средством разработки для решения поставленной задачи определена среда программирования Delphi, т.к. она обладает рядом следующих преимуществ:

1. Быстрота разработки приложения;
2. Невысокая сложность отладки приложений;
3. Наличие встроенных компонентов для доступа к Oracle,
4. Наличие встроенных визуальных компонент для работы с БД;
5. Наличие компонент Direct Oracle Access (DOA) для связи с Oracle;
6. Наличие компонент DevExpress для визуального представления информации БД.

Практически все описанные преимущества присутствуют и у среды C++ Builder. Данные две среды практически идентичны в своем функционале, что объясняется тем, что обе они разрабатываются компанией CodeGear [9].

4. Реализация системы

4.1. Выбор типа сетевой архитектуры

Трехзвенная архитектура - представляет собой дальнейшее совершенствование технологии "клиент – сервер". В трехзвенной архитектуре вся бизнес-логика (деловая логика), ранее входившая в клиентские приложения, выделяется в отдельное звено, называемое сервером приложений. При этом клиентским приложениям остается лишь пользовательский интерфейс.

В результате работа построена следующим образом:

- База данных в виде набора файлов находится на жестком диске специально выделенного компьютера (сервера сети);
- СУБД располагается на сервере сети;
- Существует специально выделенный сервер приложений, на котором располагается ПО делового анализа (бизнес-логика);
- Существует множество клиентских компьютеров, на каждом из которых установлен так называемый "тонкий клиент" – клиентское приложение, реализующее интерфейс пользователя;
- На каждом из клиентских компьютеров пользователи имеют возможность запустить приложение – тонкий клиент. Используя предоставляемый приложением пользовательский интерфейс, он инициирует обращение к ПО делового анализа, расположенному на сервере приложений;
- Сервер приложений анализирует требования пользователя и формирует запросы к БД. Для общения используется специальный язык запросов SQL, т.е. по сети от сервера приложений к серверу БД передается лишь текст запроса.
- СУБД инкапсулирует внутри себя все сведения о физической структуре БД, расположенной на сервере.
- СУБД инициирует обращения к данным, находящимся на сервере, в результате которых результат выполнения запроса копируется на сервер приложений.

- Сервер приложений возвращает результат в клиентское приложение (пользователю).
- Приложение, используя пользовательский интерфейс, отображает результат выполнения запросов.

Для реализации дипломного проекта была использована трехзвенная клиент-серверная архитектура, которая изображена на рисунке 5.

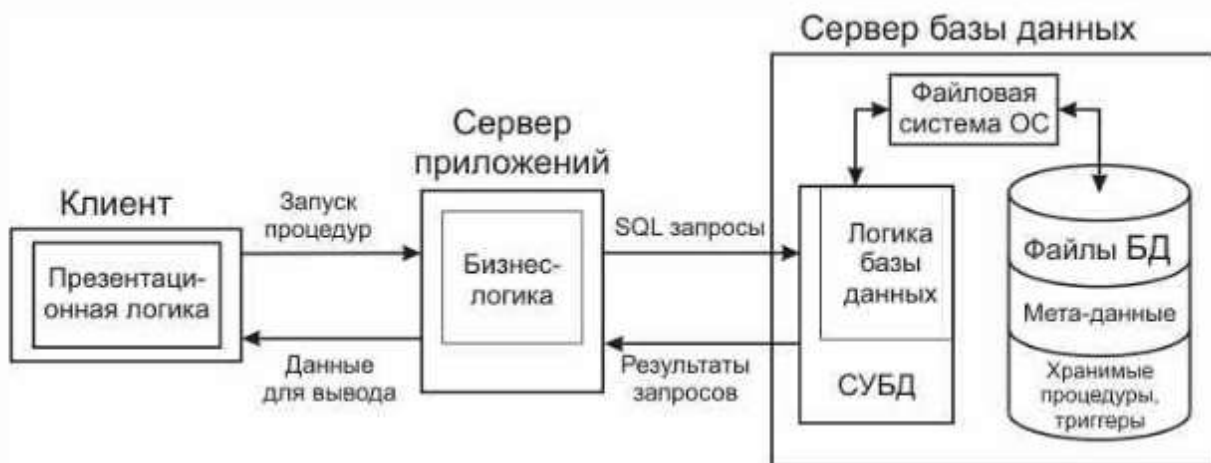


Рисунок 5 - Трёхзвенная клиент-серверная система

Целью дипломного проекта является разработать базу данных предварительной диспетчеризации обращений пользователей ООО «ИТ-СЕРВИС».

4.2. Описание ограничений целостности БД

Ограничения целостности БД задаются для того, чтобы задействовать организационные правила по данным в таблицах. Когда ограничение целостности включено, все данные в таблице должны подчиняться правилу, которое им специфицировано. Если после этого выполняется предложение SQL, которое модифицирует данные в этой таблице, то ORACLE проверяет, чтобы результирующие данные удовлетворяли ограничению целостности.

Опишем существующие ограничения ссылочной целостности в разрабатываемой БД:

- ✓ при удалении записи таблицы ДОЛЖНОСТИ в случае наличия связанных записей в таблице ИНИЦИАТОР срабатывает внутреннее исключение Oracle, происходит откат операции;

- ✓ при удалении записи таблицы ОРГАНИЗАЦИЯ в случае наличия связанных записей в таблице ДЕПАРТАМЕНТ срабатывает внутреннее исключение Oracle, происходит откат операции;
- ✓ при удалении записи таблицы ДЕПАРТАМЕНТ в случае наличия связанных записей в таблице ОТДЕЛ срабатывает внутреннее исключение Oracle, происходит откат операции;
- ✓ при удалении записи таблицы ОТДЕЛ в случае наличия связанных записей в таблице ИНИЦИАТОР срабатывает внутреннее исключение Oracle, происходит откат операции;
- ✓ при удалении записи таблицы ИНИЦИАТОР в случае наличия связанных записей в таблице ОБРАЩЕНИЕ срабатывает внутреннее исключение Oracle, происходит откат операции;
- ✓ при удалении записи таблицы ДОГОВОР в случае наличия связанных записей в таблице УСЛУГИ ДОГОВОРА срабатывает внутреннее исключение Oracle, происходит откат операции;
- ✓ при удалении записи таблицы УСЛУГА в случае наличия связанных записей в таблице УСЛУГИ ДОГОВОРА срабатывает внутреннее исключение Oracle, происходит откат операции;
- ✓ при удалении записи таблицы ИСПОЛНИТЕЛЬ в случае наличия связанных записей в таблице ОБРАЩЕНИЕ срабатывает внутреннее исключение Oracle, происходит откат операции;
- ✓ при удалении записи таблицы УСЛУГИ ДОГОВОРА в случае наличия связанных записей в таблице ОБРАЩЕНИЕ срабатывает внутреннее исключение Oracle, происходит откат операции;

Опишем существующие ограничения целостности данных в виде таблицы 3 в приложении В.

4.3. Администрирование баз данных

Администрирование любой системы управления базами данных сводится к следующему ряду задач:

1. Защита от несанкционированного доступа;
2. Администрирование пользователей и организация прав доступа клиентов к данным;
3. Архивирование и восстановление баз данных, организация графика резервного копирования.

4.3.1. Защита от несанкционированного доступа

При работе в сети компьютер нуждается в постоянной защите от вредоносных программ, таких как интернет-черви, программы-шпионы, программы, позволяющие удаленно контролировать компьютер, которые могут причинить ущерб и нарушить его работу, начиная от замедления работы и заканчивая кражей персональной информации.

Для защиты от несанкционированного доступа чаще всего используется Firewall.

Firewall – это программный или аппаратный комплекс для разграничения доступа к локальной или глобальной сети. Общий принцип его работы следующий: Firewall отслеживает все устанавливаемые соединения, после просматривает имеющийся у него список правил (заданных пользователем или администратором) и определяет, следует ли разрешить данное соединение или заблокировать.

Firewall можно разделить на следующие виды: корпоративный и персональный.

Корпоративный Firewall может быть как программным, так и аппаратным. Они устанавливаются на шлюз между локальной сетью и Интернетом (или на шлюз между двумя подсетями) и в качестве правил для проверки используется тип протокола, адрес сайта и номер порта. Правила для корпоративного Firewall задаются администратором сети.

Персональный Firewall – это программа, которая устанавливается непосредственно на компьютер обычного пользователя. Основное отличие персонального Firewall от корпоративного заключается в том, что в его правилах можно задавать не только протокол, адрес и порт для подключения, но и программы, которым разрешено это подключение устанавливать (или наоборот, принимает входящее). Почти все персональные Firewall имеют режим обучения, т.е. если для какой-либо программы не создано правила, пользователю выдается сообщение, что программа такая-то попыталась установить соединение по такому-то адресу, после чего вы должны принять решение, разрешать это соединение или нет. Соответственно, если вы выберете «разрешить», то программе будет позволено установить соединение и отправить данные. Таким образом, Firewall не является универсальным решением всех проблем безопасности, как это иногда кажется. Firewall – это инструмент, который помогает защитить компьютер от атак, но только в том случае, если вы знаете, что вы делаете.

Для того чтобы использовать Firewall правильно, можно дать следующие рекомендации:

1. Внимательно прочитывать сообщения от Firewall;
2. Предоставлять доступ только необходимым программам;
3. Предоставлять доступ только к необходимым портам, которые используются протоколами и по которым работает программа;
4. В ситуациях, в которых неясно, что делать, нужно действовать по принципу: лучше лишний раз запретить, чем лишний раз разрешить.

Определить, какие порты используются, и для каких протоколов, можно по таблице 4.

В большинстве случаев на обычный пользовательский компьютер не должны приниматься никакие входящие соединения из Интернета, кроме соединений с локального адреса 127.0.0.1. Данный ip-адрес – это адрес собственного компьютера (соединения на этот адрес могут устанавливаться различными программами-фильтрами, например, Proxomitron, или локальными

проxy-серверами, например, CoolProху) и на любой запрос о приеме входящего подключения можно отвечать «Нет» (кроме тех случаев, когда точно известно, для чего это входящее соединение нужно). Исключения могут быть в следующих случаях: FTP-клиенты могут принимать входящие соединения при работе в активном режиме, mIRC принимает входящее соединение на порт 113 при подключении к серверу, а также в некоторых ситуациях входящие подключения может принимать ICQ.

На данный момент существует большое количество персональных Firewall: Augnitum Outpost, ZoneAlarm, Kerio Personal Firewall, Kaspersky Antihacker, Ms Trend Micro, Tiny Personal Firewall и многие другие. Почти все они примерно одинаковы по функциональности, и различия заключаются в интерфейсе и удобстве использования. В Oracle используется встроенный Oracle DataBase Firewall.

Таблица 4 – Соотношение портов и протоколов

Тип программы	Протокол	Порты
DNS-клиент - определение адреса по символьному имени (SVCHOST.EXE)	DNS	53 (исходящий TCP и UDP)
Браузер (Internet Explorer, Opera, Mozilla)	HTTP, HTTPS	80, 443 (исходящий TCP)
Почтовый клиент (The Bat, Outlook, Eudora, Becky) – исходящая почта	SMTP	25 (исходящий TCP)
Почтовый клиент (The Bat, Outlook, Eudora, Becky) – входящая почта	POP3, IMAP	110 (исходящий TCP)
FTP-клиент (Windows Commander, CuteFTP, SmartFTP, BulletProof FTP)	FTP	21 (исходящий TCP), порты в диапазоне 2000- 4000 (входящие TCP)
ICQ-клиент (ICQ, &RQ, Miranda, Trillian)	ICQ	5190 (исходящий TCP)
IRC-клиент (mIRC)	IRC, Ident	6666-6669 (исходящий TCP), 113 (входящий TCP)

4.3.2. Администрирование пользователей и организация прав доступа клиентов к данным Oracle

Основное место в средствах защиты занимает учётная запись пользователя базы данных Oracle.

CREATE USER – это команда SQL, которая может использоваться для определения учётной записи Oracle в базе данных. После создания учётной записи пользователя Oracle она не может использоваться, пока пользователь не получит, по меньшей мере одну системную привилегию. Системная привилегия CREATE SESSION позволяет пользователю создавать сеанс по отношению к базе данных Oracle. Это – необходимая привилегия, которую должна иметь учётная запись пользователя, без неё учётная запись пользователя Oracle не может использоваться.

При первоначальном создании пользователя Oracle можно определить заданное по умолчанию табличное пространство, в котором будут создаваться объекты пользователя. Если заданное по умолчанию табличное пространство не определено, пользователю будет назначено табличное пространство SYSTEM в качестве заданного по умолчанию, которое будут использовать объекты базы данных. В составе оператора CREATE USER может использоваться фраза DEFAULT TABLESPACE для определения того, что объекты пользователя должны быть помещены в табличное пространство, отличное от SYSTEM. Пользователю Oracle, также должна быть назначена квота, которая определяет, сколько памяти он может использовать в табличном пространстве.

Другой способ создания пользователя состоит в том, чтобы предоставить пользователю роли CONNECT, RESOURCE и DBA. Хотя это и быстрый метод, он включён, прежде всего, для совместимости с предыдущими версиями программного обеспечения Oracle. Команда CREATE USER – более предпочтительный метод, поскольку в одной команде можно указать и квоту и другие установки.

Можно использовать команду ALTER USER для изменения таких параметров пользователя, как пароль, заданные по умолчанию временные табличные пространства и квота памяти.

Для удаления пользователя из базы данных используется команда DROP USER, которая удаляет запись пользователя из словаря данных Oracle. Если пользователь Oracle владеет какими-либо объектами базы данных, можно либо удалить каждый из объектов перед использованием команды DROP USER, либо использовать в DROP USER опцию CASCADE для автоматического уничтожения всех объектов при удалении учётной записи пользователя.

Во время работы СУБД возможны повреждения базы данных вследствие неправильной работы аппаратуры, плохого энергоснабжения, ошибок пользователей или программ, различных техногенных факторов. Поэтому необходимо регулярно создавать резервные копии баз данных и журналов транзакций. Процесс восстановления базы данных состоит в восстановлении ее с резервной копии, внесении в нее изменений из журнала транзакции, датированных временем между созданием этой копии и последним копированием журнала и, если это необходимо, ручным восстановлением изменений, внесенных в промежуток времени между последним копированием журнала транзакций и потерей информации.

Не исключено, что повреждению может подвергнуться и резервная копия. Поэтому рекомендуется хранить копии на отдельных носителях (съёмных жестких дисках, магнитной ленте и т.д.), которые не могут быть повреждены вместе с оборудованием. Кроме того, рекомендуется хранить резервные копии на несколько колен назад.

Сам процесс резервного копирования и восстановления полностью автоматизирован. Возможно даже составление расписаний резервного копирования, в соответствии с которыми эта процедура будет автоматически выполняться в определенное время для указанных баз данных.

Oracle записывает все изменения, которые вносятся в находящиеся в памяти блоки данных, в оперативные журналы повторного выполнения (online

redo logs), и делает это, как правило, перед их записью в файлы базы данных. Во время процесса восстановления Oracle использует зафиксированные в файлах этих журналов изменения для приведения базы данных в актуальное состояние. Oracle поддерживает два режима для управления такими файлами.

Режим архивирования журналов ARCHIVELOG.

В режиме ARCHIVELOG Oracle сохраняет (архивирует) заполненные журналы повторного выполнения. Следовательно, на сколько давно бы не выполнялось резервное копирование, если используется режим ARCHIVELOG, базу данных всегда можно будет восстановить до любой точки во времени с помощью архивных журналов.

Режим без архивирования журналов NOARCHIVELOG

В данном режиме заполненные журналы повторного выполнения перезаписываются, а не сохраняются. Это, следовательно, означает, что в случае использования режима NOARCHIVELOG выполнять восстановление можно будет только из резервной копии и что все остальные изменения, которые были внесены в базу данных после выполнения резервного копирования, будут утрачиваться. Этот режим обеспечивает возможность выполнения восстановления только после отказа экземпляра базы данных. В случае возникновения неполадок с носителем (например, потеря диска), функционирующую в режиме NOARCHIVELOG базу данных можно будет восстановить только из резервной копии и, естественно, с потерей всех изменений, которые были внесены в нее после создания этой резервной копии.

В режиме ARCHIVELOG функционируют почти все производственные базы данных. Режим NOARCHIVELOG используется только тогда, когда есть возможность восстановить данные из других источников, или когда база данных пока находится только на стадии разработки или тестирования и потому не нуждается в возможности восстановления ее данных с точностью до минуты.

Подвергать резервному копированию можно как всю базу данных, так и только ее часть, вроде входящего в ее состав табличного пространства или

файла данных. Обратите внимание, что в случае, когда база данных функционирует в режиме NOARCHIVELOG, осуществлять резервное копирование лишь части базы данных, также называемое частичным резервным копированием (partial database backup), нельзя, если только все те табличные пространства и файлы, которые подлежат резервному копированию, не являются доступными только для чтения. Выполнять резервное копирование всей базы данных, также называемое полным резервным копированием (whole database backup), можно как в режиме ARCHIVELOG, так и в режиме NOARCHIVELOG.

Чаще всего выполняется полное резервное копирование. Оно предполагает копирование не только всех файлов данных, но и еще одного важного файла – управляющего. Без управляющего файла Oracle не будет открывать базу данных, поэтому для восстановления помимо резервных копий всех файлов данных, необходимо также обязательно обладать и новейшей резервной копией управляющего файла.

На рисунке 6 представлена схема восстановления данных БД.

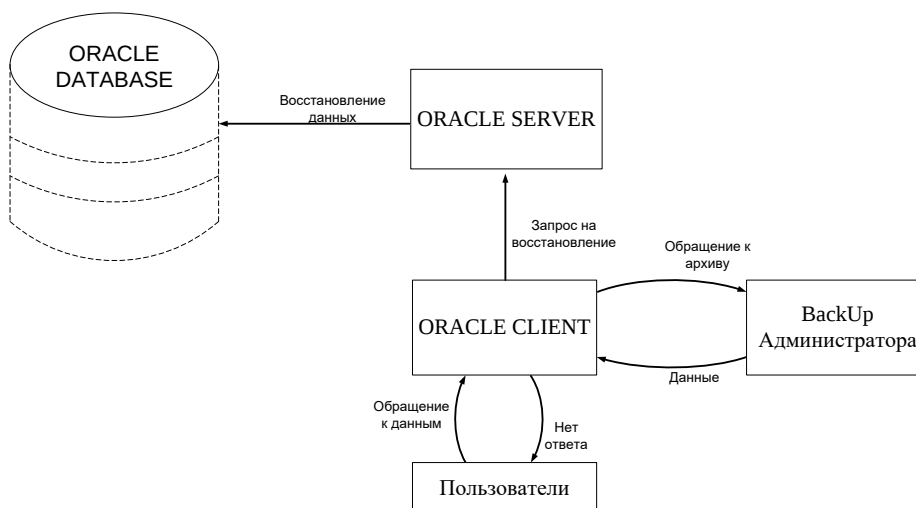


Рисунок 6 - Схема восстановления данных

4.4. Требования к составу и параметрам технических средств

Для корректной работы сервера СУБД Oracle 11g достаточно следующей конфигурации [11]:

1. Сервер базы данных, сервер приложений:
 - Процессор Intel Xeon CPU 3,0 ГГц;
 - Оперативная память (RAM) 16 Гб ;
 - Свободное место на диске HDD - 200 Гб
2. Рабочая станция:
 - Процессор Pentium IV 2,5 ГГц,
 - Оперативная память (RAM) 2 Гб,
 - Свободное место на диске HDD – 40 Гб.
 - Операционная система - Windows XP SP2 и выше
 - Устройства ввода (клавиатура, мышь);
 - Устройства вывода (монитор);
 - Локальная сеть пропускная способность – от 10 Мбит/с (оптимально 100 Мбит/с).

5. Описание приложения

5.1. Описание приложения администрирования пользователей

Для администрирования пользователей программы был разработан вспомогательный модуль приложения, позволяющий редактировать справочник исполнителей, организация и списка оказываемых услуг. Кратко опишем его работу.

На рисунке 7 изображено окно соединения с сервером Oracle, которое отображается при запуске приложения.

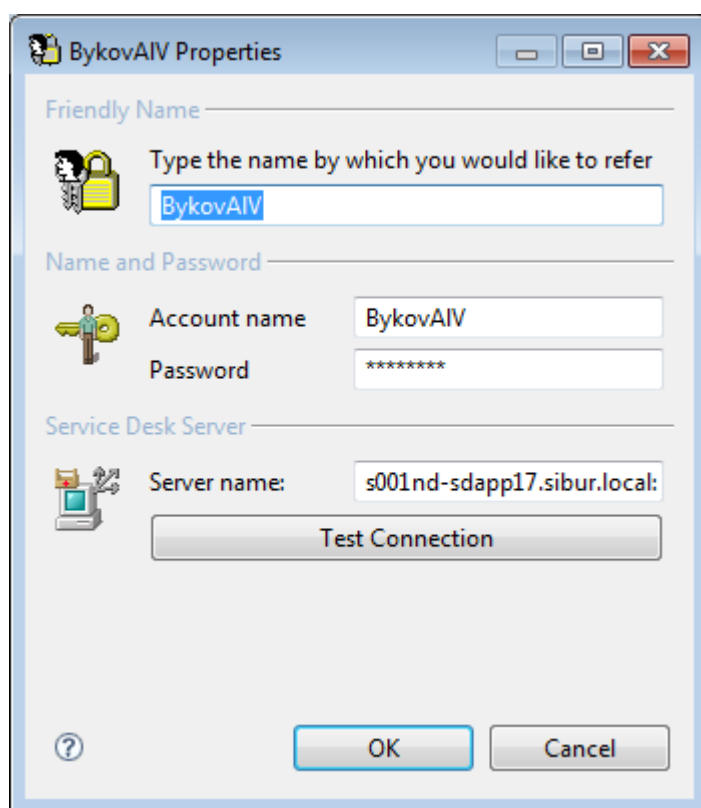


Рисунок 7 - Окно соединения с сервером Oracle

Данная форма помимо основной функции – аутентификации, также проверяет пользователя на его принадлежность к группе администраторов.

При успешном соединении с сервером БД на экране появляется главная форма приложения, которая предоставлена на рисунке 8.

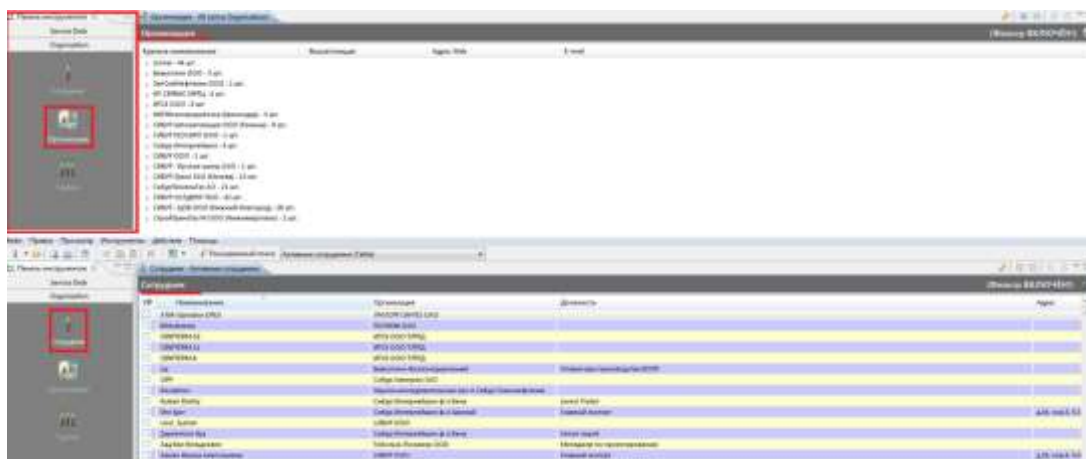


Рисунок 8 – Редактирование информации об организации и личных карточек пользователей

Данная форма позволяет создавать, редактировать и удалять записи об исполнителях и об организациях, которые входят в домен sibur.local.

Форма редактирования карточки пользователя предприятий ПАО «СИБУР Холдинг» представлена на рисунке 9, как правило, информация, которая отображается в личной карточке сотрудника, подтягивается напрямую из Active Directory.

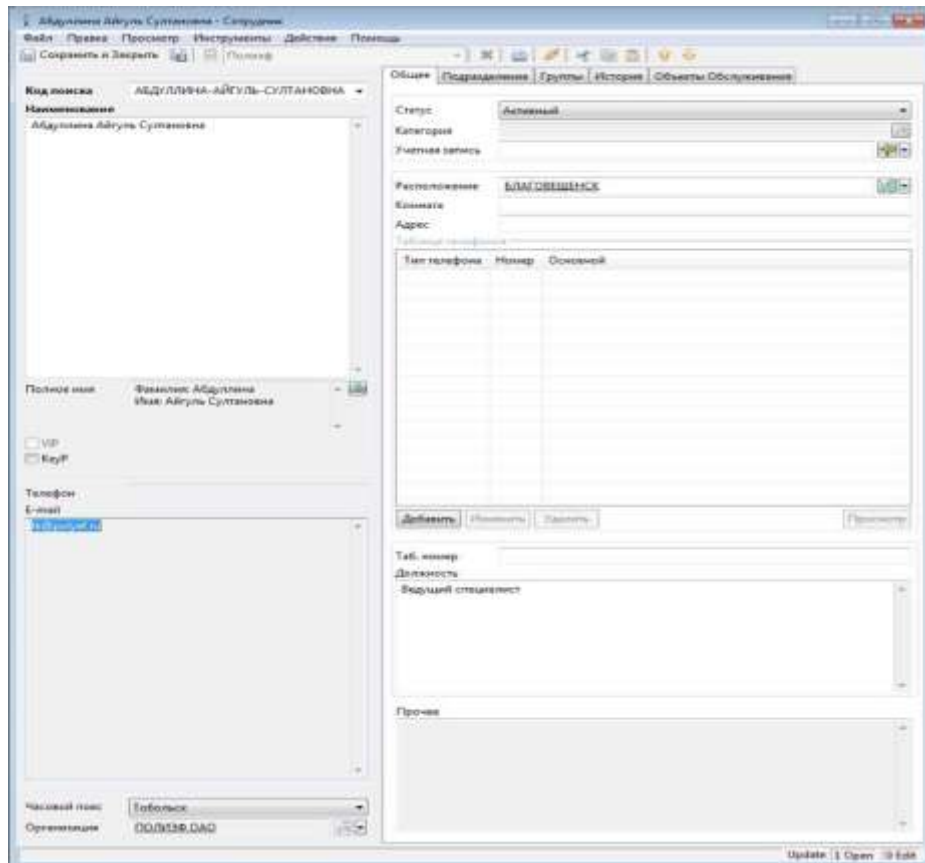


Рисунок 9 – Редактирование личной карточки сотрудника.

Форма редактирования (добавления/удаления/изменения) услуги и SLA по обращениям предоставлена на рисунке 10.

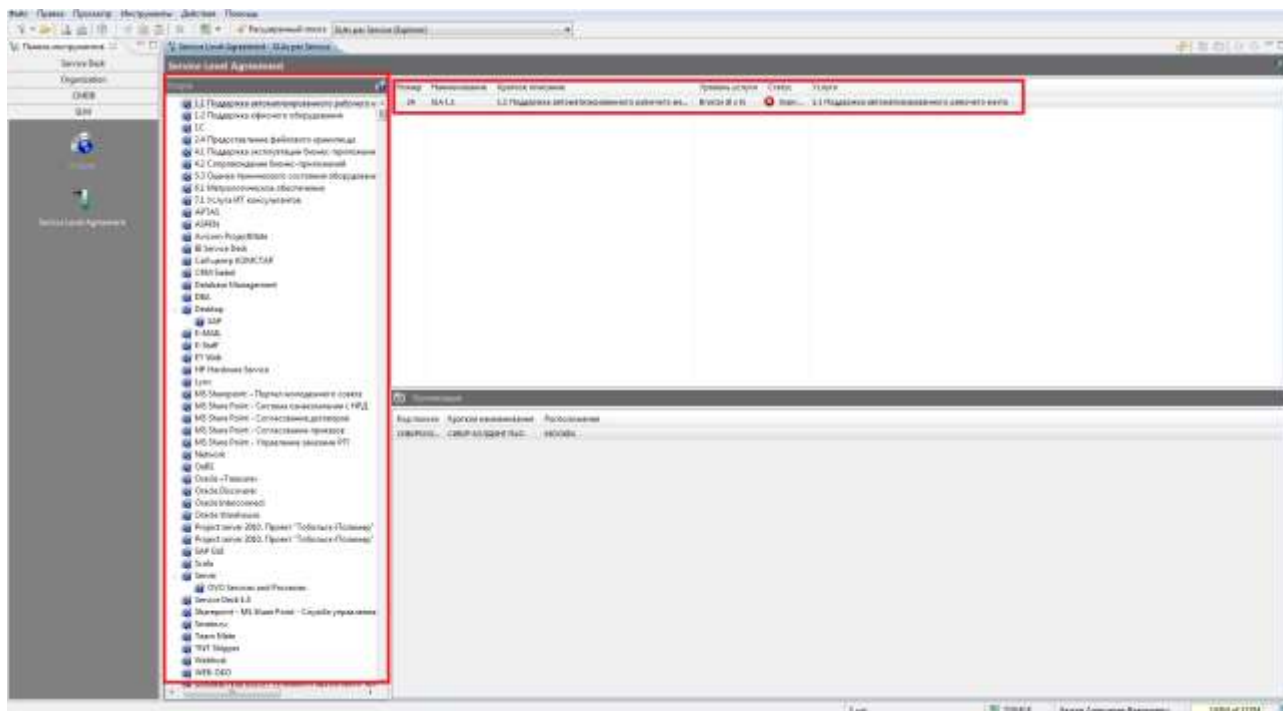


Рисунок 10 – Форма редактирования услуги и SLA

На данной форме отображены все услуги, которые в настоящее время оказываются компанией по всему Холдингу. Можно открыть любую услугу и внести корректировки, а именно подписать новую организацию на выбранную услугу или изменить метрики по расчету крайнего срока.

5.2. Описание клиентской части приложения

При запуске приложения пользователь должен ввести свои логин и пароль для инициализации, окно соединения идентичное, как на рисунке 7. Разница состоит в том, что при аутентификации учетной записи пользователя проверяется проверка с какими права пользователь зайдет в систему.

В зависимости от типа пользователя (пользователь/администратор) программа может блокировать доступ к определенным задачам или модулям в системе. При успешной авторизации пользователя на рисунке 11 показана основная форма главного окна приложения.

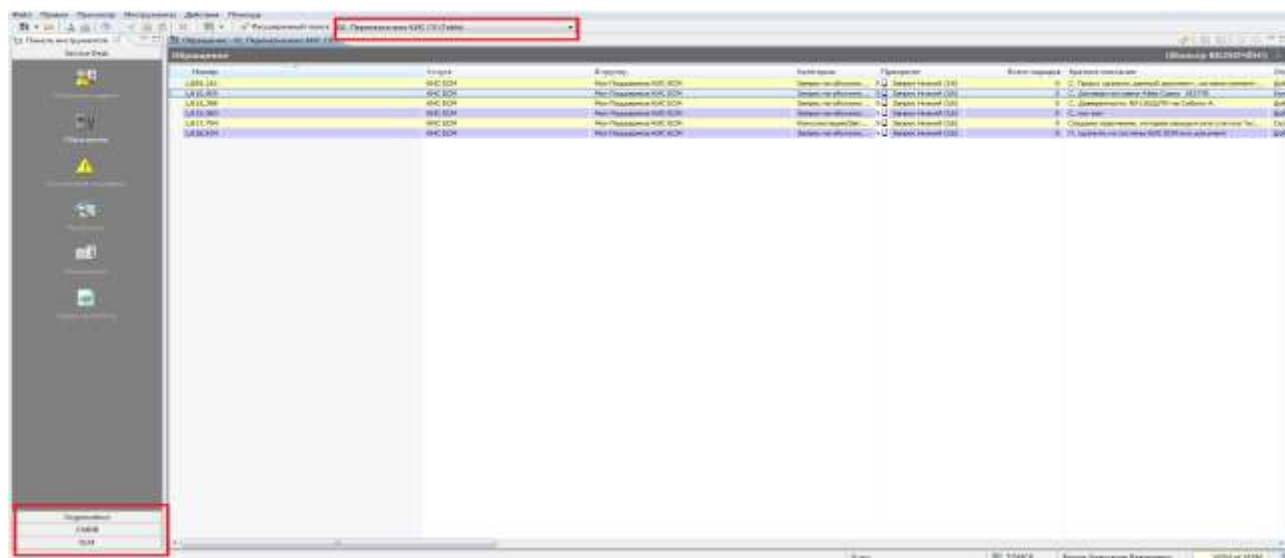


Рисунок 11 – Основная форма приложения

Доступ к закрытым задачам у обычного пользователя показан в нижнем левом углу, также в области статуса отображается ORACLE-имя пользователя.

В верхней части экрана выделено необходимо представление для работы, также был проведен опрос у специалистов 1 линии на тему, какие необходимо создать представления, которые часто всего используются в работе. Выбрать необходимое преставление можно с помощью выплывающего списка, который расположен в верхней части формы.

На рисунке 12 изображена форма задачи “Обращение”.

Рисунок 12 – форма «Обращение»

В верхней части формы задачи «Обращение» расположена панель ID пользователя. Данные пользователя подтягиваются из Active Directory, если пользователя нет в БД (такой случай возможен из-за срочного создания УЗ пользователя в ИС ЕСУИД, следовательно в разрабатываемый продукт не сразу подтягивается информация о новой УЗ), то в поле «Инициатор» можно в ручном режиме выбрать «Анонимного пользователя» того предприятия к которой он относится. В центральной части формы «Обращение» указана «Услуга» согласно выбранной услуге проставляется крайний срок, который рассчитывается согласно регламенту SLA. Также в верхней части формы «Обращение» указаны «Влияние» и «Категория», которые напрямую влияют на крайний срок по обращению подробно о расчёте метрики описано в разделе 2.1.

Для необходимости привлечения дополнительных специалистов смежных функциональных групп для решения обращения с помощью наряда или изменения/проблемы добавлены в верхней части формы вкладки «Наряды» и соответственно «Связанные решения». С помощью данных вкладок можно привлечь смежную функциональную группу. Пример вкладки «Связанные решения» привязка к обращениям новой проблемы. Форма «Проблема» предоставлена на рисунке 13. При закрытии проблемы все привязанные к ней обращения, автоматически переводятся в статус «Выполнено», что естественно сокращает затраты времени при повторных просмотрах актуального статуса.

Рисунок 13 – Форма «Проблема»

6. Финансовый менеджмент, ресурсо-эффективность и ресурсосбережение

Данная работа посвящена разработке системы, а именно нового программного продукта «База данных для предварительной диспетчеризации пользователей ООО ИТ-СЕРВИС». Данный проект, как и любой разрабатываемый проект, вне зависимости от того какой характер он включает в себя, научный или практический в конечном итоге проделанная работа имеет экономическую составляющую, которую необходимо оценить с целью предоставления полной картины значимости осуществленной работы и ее вклада в заданную предметную область.

Целью экономического раздела является оценка и предварительный анализ экономических аспектов разрабатываемого проекта, необходимость обоснования экономической эффективности нового (разрабатываемого) продукта.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Выбор аналога сравнения;
2. Необходимость замены аналога на разрабатываемый продукт;
3. Оценка эксплуатационно-технического уровня нового продукта, по сравнению аналогом;
4. Организация и планирование комплекса работ по дипломному проекту;
5. Определение финансовых затрат на разработку дипломного проекта;
6. Расчет стоимости опытного образца и аналога;
7. Затраты на монтаж опытного образца;
8. Расчет капитальных вложений;
9. Расчет эксплуатационных затрат;
10. Оценка финансово-экономической эффективности нового продукта.

6.1 Выбор аналога сравнения и необходимость замены на разрабатываемый продукт

Во время проектирования нового программного продукта в качестве аналога была выбрана уже существующая программная система, которая в настоящее время внедрена в промышленную эксплуатацию в компании ООО «ИТ-СЕРВИС». Название программной системы «**OmniTracker**», а на роль второго аналога взята система «**MOSAIC**»

Новый разработанный продукт «База данных предварительной диспетчеризации пользователей ООО ИТ-СЕРВИС» имеет по сравнению с аналогом улучшенные технические характеристики, новый интерфейс, новый подход в сетевой архитектуре. Вся необходимая информация отображена в таблице 5, которая расположена в приложении Г.

Также разрабатываемый продукт имеет следующее и, наверное, самые основные улучшения и превосходства нового продукта:

- уменьшение затрат на распространение лицензии;
- увеличение быстродействия системы;
- изменения интерфейса для пользователей системы.

6.2. Организация и планирование работы

Одной из составляющих успешной реализации проекта служит рациональное планирование занятости каждого из его участников, а также определение сроков выполнения определенных этапов работы над проектом.

В данном разделе приводится перечень этапов работы, исполнителей, а также оценивается степень участия каждого из участников в том или ином этапе. Данные по перечню работ и продолжительности работ представлены в таблице 6. Число исполнителей данного проекта равно двум – непосредственный исполнитель и научный руководитель.

Таблица 6 – Перечень работ и продолжительность их выполнения

Этапы работы	Исполнители	Загрузка исполнителей
<i>постановка задачи</i>		
1. Постановка задачи, определение целей	НР	НР - 100 %
<i>анализ</i>		
2. Выявлений требований к программе	НР, И	НР – 100 %, И – 30%
3. Подбор и обзор литературы, обзор существующих решений	НР, И	НР – 30%, И – 100%
4. Календарное планирование	НР, И	НР – 100%, И – 10 %
<i>проектирование</i>		
5. Разработка модели системы	НР, И	НР – 80 % , И – 100%
6. Разработка алгоритма реализации	НР, И	НР – 70 %, И – 100%
<i>реализация</i>		
7. Разработка приложения	И	И – 100%
<i>тестирование</i>		
8. Тестирование	НР, И	НР – 10%, И – 100%
9. Анализ результата	И	И – 100%
<i>документирование</i>		
10. Расчет экономических показателей	И	И – 100%
11. Оценка показателей безопасности жизнедеятельности	И	И – 100%
12. Оформление пояснительной записки	И	И – 100%
13. Подведение итогов	НР, И	НР – 50%, И – 100%

6.2.1. Продолжительность этапов работ

Планирование работ по магистерской диссертации заключается в составлении перечня работ, необходимых для достижения поставленных задач, определении исполнителей каждой работы, установлении продолжительности работ в рабочих днях, построении линейного или календарного графика. Прежде всего, необходимо правильно установить оптимальную величину работ по теме.

Процесс создания и выпуска нового программного продукта состоит из ряда этапов, выполняемыми разными исполнителями. Этот процесс должен быть скоординирован во времени. Планирование комплексных работ выражается в составлении календарных планов.

Все этапы разработки должны быть упорядочены во времени. Необходимо определить последовательность работ, являющуюся наиболее рациональной с точки зрения минимальных затрат времени на осуществления всего комплекса работ по созданию устройства.

В данном разделе осуществляется расчет продолжительности работ с использованием опытно-статистического метода. Данный метод имеет 2 способа реализации - аналоговый и экспертный. Так как аналоговый способ подразумевает наличие некоторого идентичного выполняемой научно-исследовательской работе (по всем значительным параметрам) проекта, то, в условиях отсутствия последнего, принято решение применять экспертный способ. Расчет продолжительности этапов работ приведен в таблице 7. Для построения таблицы используются следующие параметры:

- Ожидаемые (вероятные) значения продолжительности работ ($t_{ож}$):

$$t_{ож} = \frac{3 \cdot t_{min} + 2 \cdot t_{max}}{5} \quad (1),$$

где

t_{min} – минимальная продолжительность работы, дни;

t_{max} – максимальная продолжительность работы, дни;

- Продолжительность выполнения каждого этапа в рабочих днях ($T_{РД}$):

$$T_{РД} = \frac{t_{ож}}{K_{вн}} \cdot K_{д} \quad (2),$$

где

$t_{ож}$ – продолжительность работы, дни;

$K_{вн}$ – коэффициент выполнения работ с учетом внешних факторов, влияющих на соблюдение предварительно определенных длительностей работы (в данном случае используется $K_{вн} = 1$);

$K_{д}$ – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ (принимает значения в интервале $[1; 1,2]$, для расчетов используется величина 1,2 с целью максимально учесть непредвиденные временные затраты).

- Продолжительность выполнения этапа в календарных днях ($T_{КД}$):

$$T_{КД} = T_{РД} \cdot T_K \quad (3),$$

где

T_K – коэффициент календарности, служит для перехода от длительности работ в рабочих днях к их аналогам в календарных днях. Данный коэффициент имеет следующую формулу:

$$T_K = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}} \quad (4), \text{ где}$$

$T_{КАЛ}$ – календарные дни (365 дней)

$T_{ВД}$ – выходные дни (52 или 104 дня, для шестидневной и пятидневной рабочей недели соответственно)

$T_{ПД}$ – праздничные дни (10 дней)

Таким образом, для шестидневной рабочей недели получаем следующий коэффициент календарности:

$$T_K = \frac{365}{365 - 52 - 10} = 1,205$$

Таблица 7 – Трудозатраты на выполнения проекта

Этап	Исполнители	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.-дн.			
					$T_{РД}$		$T_{КД}$	
		t_{min}	t_{max}	$t_{ож}$	НР	И	НР	И
1	2	3	4	5	6	7	8	9
Постановка задачи, определение	НР	2	4	2,8	3,36	-	4,05	-
Выявление требований к программе	НР, И	2	3	2,4	2,88	0,86	3,47	1,04
Подбор и обзор литературы, обзор существующих решений	НР, И	10	14	11,6	4,18	13,92	5,03	16,77
Календарное планирование	НР, И	2	4	2,8	3,36	0,34	4,05	0,4
Разработка модели системы	НР, И	15	20	17	16,32	20,40	19,66	24,57
Разработка алгоритма реализации	НР, И	20	25	22	18,48	26,40	22,26	31,80
Разработка приложения	И	125	150	135	-	162,00	-	195,15
Тестирование	НР, И	1	2	1,4	0,17	1,68	0,20	2,02
Анализ результатов	И	10	15	12	-	14,40	-	17,35
Расчет экономических показателей	И	2	4	2,8	-	3,36	-	4,05
Оценка показателей безопасности жизнедеятельности	И	2	4	2,8	-	3,36	-	4,05
Оформление пояснительной записки	И	7	14	9,8	-	11,76	-	14,17
Подведение итогов	НР, И	3	5	3,8	2,28	4,56	2,75	5,49
Итого				226,2	51,02	263,04	61,46	316,86

На рисунке 14 представлена диаграмма Ганта для исполнителя проекта в календарных днях.

Диаграмма Ганта

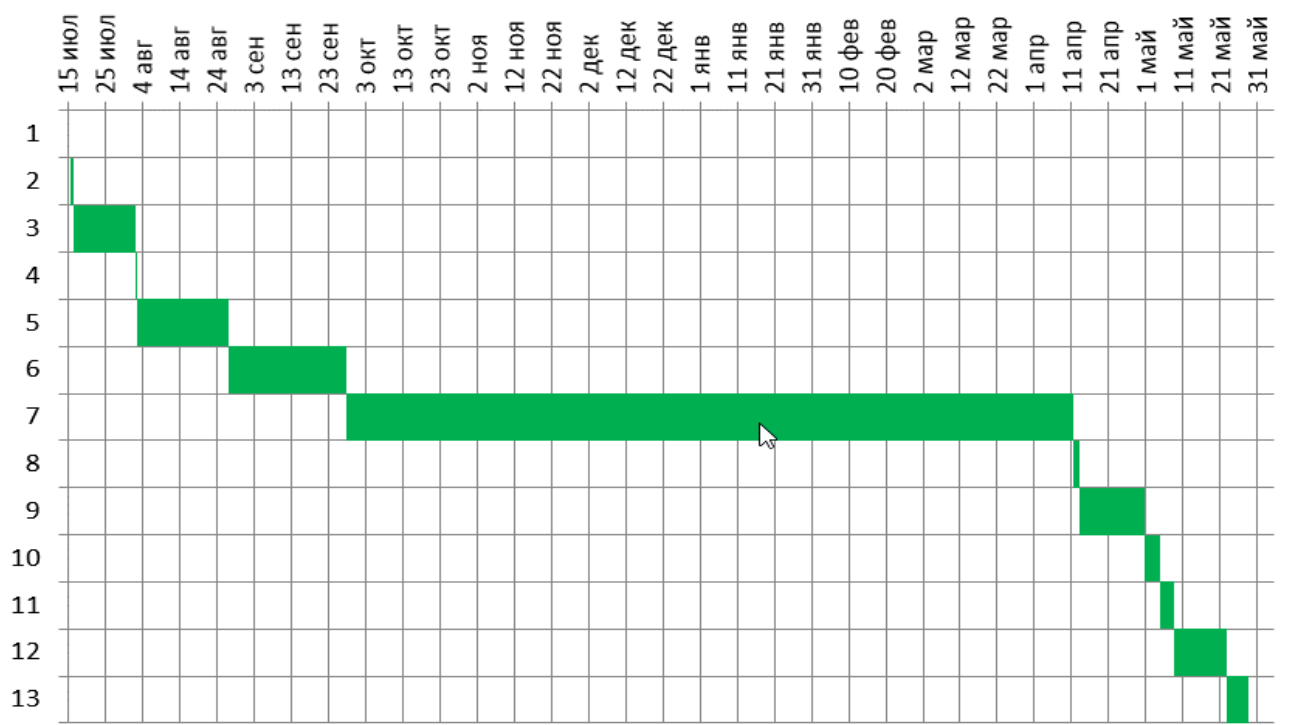


Рисунок 14 – Диаграмма Ганта для исполнителя проекта

6.2.2 Расчет накопления готовности проекта

В данном разделе производится оценка текущих состояний работы над проектом. Величина накопления готовности работы показывает, на сколько процентов по окончании текущего (*i*-го) этапа выполнен общий объем работ по проекту в целом. Нарастание технической готовности работы и удельный вес каждого этапа представлены в таблице 8.

Для определения степени готовности проекта используется следующая формула:

$$CG_i = \frac{TP_i^H}{TP_{общ.}} = \frac{\sum_{k=1}^i TP_k}{TP_{общ.}} = \frac{\sum_{k=1}^i \sum_{j=1}^m TP_{km}}{\sum_{k=1}^I \sum_{j=1}^m TP_{km}} \quad (5), \text{ где}$$

$TP_{общ.}$ – общая трудоемкость проекта;

$TP_i(TP_k)$ – трудоемкость *i*-го (*k*-го) этапа проекта;

TP_i^H – накопленная трудоемкость *i*-го этапа проекта по его завершении

TP_{ij} (TP_{kj}) – трудоемкость работ, выполняемых j-м участником на i-м этапе, здесь $j = \overline{1, m}$ – индекс исполнителя (в данном случае $m = 2$, так как в проекте 2 исполнителя). Расчет данной величины производится на основании столбцов 6 и 7.

Таблица 8 – Нарастание технической готовности работы и удельный вес каждого этапа

№ п/п	Этап	ТР _i , %	СГ _i , %
1	Постановка задачи, определение целей	1,07	1,07
2	Выявлений требований к программе	1,19	2,26
3	Подбор и обзор литературы, обзор существующих решений	5,76	8,02
4	Календарное планирование	1,18	9,20
5	Разработка модели системы	11,69	20,89
6	Разработка алгоритма реализации	14,29	35,18
7	Разработка приложения	51,58	86,76
8	Тестирование	0,59	87,35
9	Анализ результатов	4,59	91,94
10	Расчет экономических показателей	1,07	93,01
11	Оценка показателей безопасности жизнедеятельности	1,07	94,08
12	Оформление пояснительной записки	3,74	97,82
13	Подведение итогов	2,18	100

6.3 Расчет сметы затрат на выполнение проекта

С учетом специфики проделанной работы для рассматриваемого проекта производится оценка следующих расходов:

- заработная плата;
- социальный налог;
- расходы на электроэнергию (без освещения);
- амортизационные отчисления;
- арендная плата за пользование имуществом;
- прочие расходы.

Ввиду отсутствия привлечения помощи сторонних организаций, командировок, привлечения консалтинговых услуг, расходы по соответствующим статьям не оцениваются применительно к данной работе.

Помимо вышеперечисленного, не рассматриваются также затраты на материалы ввиду их незначительности в масштабе данной работы.

6.3.1. Расчет заработной платы

В данном разделе расчет основной заработной платы производится на основе величины месячного оклада исполнителя и трудоемкости каждого этапа.

Расчет затрат на заработную плату представлен в таблице 9. Для расчета данной таблицы использовались следующие параметры:

- Месячный оклад исполнителей проекта (МО)

В данном случае рассматриваются следующие оклады:

1) научный руководитель: должность – доцент, степень – кандидат технических наук;

2) исполнитель: младший научный сотрудник, степень – нет (оклад согласно месту прохождения преддипломной практики).

- Среднедневная тарифная заработная плата ($ЗП_{дн-т}$), рассчитывается по формуле 6:

$$ЗП_{дн-т} = \frac{МО}{24,83} \quad (6), \text{ где}$$

МО – месячный оклад исполнителя,

Значение 24, 83 – количество рабочих дней при шестидневной рабочей неделе при условии 298 рабочих дней в году.

- Интегральный коэффициент ($K_{и}$) – служит для перехода от базовой суммы заработной платы исполнителя, связанной с участием в проекте к полному заработку. Рассчитывается по формуле 7:

$$K_{и} = K_{пр} \cdot K_{допЗП} \cdot K_{р} \quad (7),$$

где

$K_{пр}$ – премиальный коэффициент ($K_{пр} = 1,1$);

$K_{допЗП}$ – коэффициент дополнительной заработной платы ($K_{допЗП} = 1,188$);

$K_{р}$ – коэффициент районной надбавки ($K_{р} = 1,3$).

Таким образом, получаем следующий интегральный коэффициент:

$$K_{\text{и}} = 1,1 \cdot 1,188 \cdot 1,3 = 1,699$$

Таблица 9 – Затраты на заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./раб.день	Затраты времени, раб. дни	Коэф-т	Фонд з/платы, руб.
НР	23 264, 86	936,97	51	1,699	81 179,50
И	14 874, 45	599,05	263	1,699	267 653,19
Итого:					348 2,69

6.3.2 Расчет затрат на социальный налог

Затраты на единый социальный налог (ЕСН) включают в себя следующие отчисления:

- пенсионный фонд;
- социальное страхование;
- медицинское страхование.

ЕСН составляет 30% от полной заработной платы по проекту:

$$C_{\text{соц}} = C_{\text{зп}} \cdot 0,3 \quad (8)$$

Таким образом, для разрабатываемого проекта получаем:

$$C_{\text{соц}} = 348\,832,69 \cdot 0,3 = 104\,649,8 \text{ руб.}$$

6.3.3. Расчет затрат на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию, используемую оборудованием в ходе выполнения проекта и рассчитывается по формуле 9:

$$C_{\text{эл.об}} = P_{\text{об}} \cdot t_{\text{об}} \cdot C_{\text{э}} \quad (9),$$

где

$P_{\text{об}}$ – мощность, потребляемая оборудованием, кВт;

$t_{\text{об}}$ – время работы оборудования, час;

$C_{\text{э}}$ – тариф на 1кВт·час.

Значения параметров

1) $C_{\text{э}}$ для ТПУ составляет 5,257 руб/ кВт·час (с НДС)

2) $t_{\text{об}} = T_{\text{рд}} \cdot K_t$, где

- $T_{\text{рд}}$ – трудозотраты исполнителя из расчета на 8 часовой рабочий день;

- $K_t \leq 1$ – коэффициент использования оборудования по времени (в данном случае используется значение 0,9).

3) $P_{\text{об}} = P_{\text{ном}} \cdot K_c$ (10), где

- $P_{\text{ном}}$ – номинальная мощность оборудования, кВт;

- $K_c \leq 1$ – коэффициент загрузки, зависит от средней степени использования номинальной мощности. Для данного проекта используется значение 1.

Расчет затрат электроэнергию представлен в таблице 10

Таблица 10 – Затраты на электроэнергию технологическую

Наименование оборудования	Время работы оборудования $t_{\text{об}}$, час	Потребляемая мощность $P_{\text{об}}$, кВт	Тариф $C_{\text{э}}$, руб/ кВт·час	Затраты $\text{Э}_{\text{об}}$, руб.
Персональный компьютер	$8 \cdot 263 \cdot 0,9 = 1893$	0,3	5,257	2986,39
Персональный компьютер	$8 \cdot 51 \cdot 0,9 = 367$	0,3	5,257	579.11
Сервер непрерывной интеграции	$24 \cdot 316 \cdot 0,5$	0,75	5,257	14950.90
Лазерный принтер	10	0,1	5,257	5,26
Итого				18521,66

6.3.4. Расчет амортизационных расходов

Для расчета амортизационных расходов используется следующая формула 11:

$$C_{\text{ам}} = \frac{N_{\text{д}} \cdot C_{\text{об}} \cdot t_{\text{рф}} \cdot n}{F_{\text{д}}} \quad (11),$$

где

N_A – годовая норма амортизации единицы оборудования;

C_{OB} – балансовая стоимость единицы оборудования;

$t_{рф}$ – фактическое время работы оборудования в ходе выполнения проекта;

n – число единиц задействованного оборудования;

F_d – действительный годовой фонд времени работы соответствующего оборудования.

Расчет амортизационных расходов представлен в таблице 11. Параметры расчета определяются следующим образом:

1) годовая норма амортизации единицы оборудования N_A

Величина N_A определяется как обратное значение величины CA , где CA – рамочные значения сроков амортизации $\equiv CA$ согласно постановлению правительства РФ «О классификации основных средств, включенных в амортизационные группы».

2) балансовая стоимость единицы оборудования C_{OB}

Значение взято исходя из фактической стоимости ПК и принтера на момент разработки проекта.

3) фактическое время работы оборудования в ходе выполнения проекта $t_{рф}$

4) действительный годовой фонд времени работы соответствующего оборудования F_d .

Величина берется на основании фактического режима использования оборудования в текущем календарном году

- $t_{рф}$: для ПК – 2104 и 408, для принтера – 10, для сервера – 7584

Таблица 11 - Амортизационные расходы

	H_A	$C_{об}$, руб.	$t_{рф}$, часов	n , шт	F_d , час	$C_{ам}$, руб
ПК	0,4	60 000	2104	1	2384	21181,21
ПК	0,4	60 000	408	1	2384	4107,38
Сервер	0,4	300 000	7584	1	8640	131666,67
принтер	0,5	15 000	10	1	596	125,84
Итого						157081, 10

6.3.5. Расчет прочих расходов

В данном разделе производится оценка расходов на выполнение проекта, которые не были учтены в предыдущих статьях. Величина прочих расходов составляет 10% от суммы всех предыдущих расходов. Таким образом расчет проводится по следующей формуле 12:

$$C_{\text{проч.}} = (C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об.}} + C_{\text{ам}}) \cdot 0,1 \quad (12)$$

Для данного проекта получаем:

$$C_{\text{проч.}} = (348\,832,69 + 104\,649,80 + 18\,521,66 + 157\,081,10) \cdot 0,1 = 62\,908,52 \text{ руб.}$$

6.3.6. Расчет общей себестоимости разработки

Общая себестоимость разработки представляет суммарное значение затрат по всем статьям сметы затрат на разработку. Расчет общей себестоимости разработки представлен в таблице 12

Таблица 12 - Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Основная заработная плата	$C_{зп}$	348 832,69
Отчисления в социальные фонды	$C_{соц}$	104 649,80
Расходы на электроэнергию	$C_{эл.}$	18 521,66
Амортизационные отчисления	$C_{ам}$	157 081,10
Прочие расходы	$C_{проч}$	62 908,52
Итого		691 993, 77

6.3.7. Расчет прибыли

Одним из вариантов приблизительной оценки прибыли является определение ее размера в пределах от 5 % до 20% от полной себестоимости проекта. Приведем для данной проекта расчет минимальной и максимальной прибыли исходя из заданных пределов.

Таким образом:

$$\text{Максимальная прибыль} = 0,2 \cdot 691\,993,77 = 138\,398,75 \text{ руб.}$$

$$\text{Минимальная прибыль} = 0,05 \cdot 691\,993,77 = 34\,599,68 \text{ руб.}$$

6.3.8. Расчет НДС

НДС составляет 18% от суммы затрат на разработку и прибыли. Таким образом, для данного проекта получаем:

$$\text{НДС}_{\min} = (691\,993,77 + 34\,599,68) = 130\,786,82 \text{ руб.}$$

$$\text{НДС}_{\max} = (691\,993,77 + 138\,398,75) = 149\,470,65 \text{ руб.}$$

6.3.9. Цена разработки НИР

Данный параметр представляет сумму полной себестоимости, прибыли и НДС. Таким образом, получаем:

$$\text{Ц}_{\text{НИР_МИН}} = (691\,993,77 + 34\,599,68 + 130\,786,82) = 857\,380,28 \text{ руб.}$$

$$\text{Ц}_{\text{НИР_МАКС}} = (691\,993,77 + 138\,398,75 + 149\,470,65) = 979\,863,17 \text{ руб.}$$

6.4. Оценка экономической эффективности проекта

Экономический эффект от внедрения разработанного проекта в количественном отношении выразить сложно. Для этого требуется провести специальное трудоемкое исследование, что невозможно в рамках данной работы, поскольку, во-первых, предполагает объём равный всей дипломной работе, во-вторых, исполнитель не обладает компетенцией для выражения полного экономического эффекта в количественном отношении. Однако в качественном отношении внедрение разработанной системы окажет влияние на многие аспекты производственного процесса. Главный эффект заключается в том, что будет создана «Единая точка входа» в технической поддержки, т.е. интеграция со всеми необходимыми автоматизированными системами.

В ходе проделанных расчётов невооруженным взглядом замечено, что из расчета экономического эффекта, мы получаем сразу экономию в первый же год использования разрабатываемой базы данных с клиентским приложением к ней. Это обусловлено тем, что основные затраты, связанные с разработкой базы данных и клиентского приложения к ней отражаются в покупке лицензий на программное обеспечение, необходимое для разработки. При чем для внедрения данной системы управления базой данных на предприятии ООО «ИТ-СЕРВИС» не имеет значение количество необходимых лицензий при условии наличия отлаженного продукта. Чем больше потребуется лицензий, тем ощутимее будет видна разница в экономии, в сравнении с аналогом «OmniTracker». Все дальнейшие расходы будут связаны исключительно с поддержанием работоспособности разрабатываемой базы данных и её обслуживанием.

Для ООО «ИТ-СЕРВИС» внедрение данной системы экономически обосновано, т.к. имея свой продукт для хранения данных обращений пользователей, не требуется отдавать разработчику деньги за обслуживание.

6.5. Оценка научно-технического уровня НИР

Научно-технический уровень характеризует влияние проекта на уровень и динамику обеспечения научно-технического прогресса в данной области. Для данной оценки используется метод балльных оценок, сущность которого заключается в присвоении каждому из показателей НИР определенного количества баллов по соответствующей для данного показателя шкале.

Научно-технический уровень определяется на основании его интегрального показателя, который выражается следующей формулой:

$$I_{\text{НТУ}} = \sum_{i=1}^3 R_i \cdot n_i \quad (13),$$

где

$I_{\text{НТУ}}$ - интегральный индекс научно-технического уровня;

R_i – весовой коэффициент i -го признака научно-технического эффекта;

n_i – количественная оценка i -го признака научно-технического эффекта в баллах;

Оценка научно-технического уровня представлена в таблице 13.

Таблица 13 – Оценка научно технического уровня НИР

Значимость	Фактор НТУ	Уровень фактора	Выбранный балл	Обоснование выбранного балла
0,4	Уровень новизны	Принципиально новая	8	Новый способ организации информационной инфраструктуры, новый стандарт передачи данных
0,1	Теоретический уровень	Разработка способа	6	Разработка нового способа объединения информации
0,5	Возможность реализации	В течение первых лет	10	Реализуется на основе хорошо известных технологий

Таким образом, для данного проекта получаем следующий показатель научно-технического уровня:

$$I_{НТУ} = 0,4 \cdot 8 + 0,1 \cdot 6 + 0,5 \cdot 10 = 3,2 + 0,6 + 5 = 8,8$$

В таблице 14 приводится оценка качественных уровней НИР.

Таблица 14 - Качественная оценка показателей НИР

Уровень НТЭ	Показатель НТЭ
Низкий	1-4
Средний	4-7
Высокий	8-10

На основании таблицы 14 данная работа относится к высокому уровню научно технического эффекта. Высокий уровень НТЭ обусловлен тем, что данная работа представляет принципиально новый подход в организации информационной инфраструктуры.

В данном разделе была произведена оценка различных экономических аспектов разработанного проекта. На основании полученных результатов проекту можно дать следующую характеристику:

- по времени разработки (10 месяцев) проект относится к краткосрочному виду проектов;
- по масштабности (с учетом себестоимости и времени разработки) проект можно отнести к категории малых проектов;
- наличие новизны метода и актуальность решаемой проблемы повышают конкурентоспособность проекта в заданной предметной области.

7. Социальная ответственность

Основные работы, проводимые в рамках выполнения ВКР, были связаны с теоретическими исследованиями, анализом и структуризацией полученных данных, проектированием и программированием информационных систем. По этой причине, основными инструментами для достижения поставленной цели являются средства вычислительной техники: персональные компьютеры, сервера, периферические устройства. Однако, использование средств вычислительной техники, накладывает целый ряд вредных факторов на человека, что впоследствии снижает производительность его труда и может привести к существенным проблемам со здоровьем сотрудника.

База данных предварительной диспетчеризации обращений пользователей ООО «ИТ-СЕРВИС» - это автоматизированная информационная система. Данное программное обеспечение предназначено для обслуживания пользователей компании ПАО «СИБУР-Холдинг». Разрабатываемая автоматизированная информационная система позволяет пользователю обратиться с проблемой связанной в сфере ИТ-услуге и связи в техническую поддержку ЕКДС и знать, что данную проблему будут выполнять специалисты ответственные за данный сервис (т.е. создание единой точки входа для пользователя), также разработка использует электронные БД для составления статистической отчетности, что можно считать работой в офисном помещении. По этой причине целью данного приложения является исследование условий труда сотрудников работающих в офисных помещениях, влияния вредных факторов и некоторых (возможных) чрезвычайных ситуаций на процесс работы сотрудников.

7.1. Производственная безопасность

7.1.1. Описание рабочего места

Для анализа опасностей выбрано конкретное помещение в компании ООО «Томскнефтехим». В офисе работают 6 человек. Для работы используется современное компьютерное оборудование, соответствующее международным требованиям безопасности. В качестве мониторов используются широкоформатные ЖК мониторы.

Размер кабинета, где проводились работы, имеет следующие параметры:

- Длина комнаты $A = 6,3$ м;
- Ширина комнаты $B = 5,25$ м;
- Высота комнаты $H = 3,54$ м;
- Площадь комнаты $S = A \cdot B = 6,3 \cdot 5,25 = 33$ м²;
- Объем комнаты $V = S \cdot H = 33 \cdot 3,54 = 117$ м³.

Исходя из полученных результатов можно заметить, что на каждого сотрудника приходится 19,5 м³ объема помещения и 5,5 м² площади, что в результате удовлетворяет требованиям санитарных норм [15], согласно которым для одного сотрудника, работающего на ЖК-мониторе должна быть предусмотрена площадь величиной не менее 4,5 м² с учетом максимального числа одновременно работающих в смену.

Определим требования нормативных актов к выявленным вредным факторам и опишем организационные мероприятия и технические средства, предотвращающие или уменьшающие воздействие вредных производственных факторов на сотрудников данного предприятия.

7.1.2. Микроклимат

Микроклимат производственных помещений – климат внутренней среды этих помещений, который определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха.

Особенно большое влияние на микроклимат, оказывают источники теплоты, находящиеся в помещении. Основными источниками теплоты в

описываемом помещении являются: ЭВМ, вспомогательное оборудование, приборы освещения, обслуживающий персонал.

В санитарных нормах установлены величины параметров микроклимата, создающие комфортные условия. Нормы устанавливаются в зависимости от времени года, характера трудового процесса и характера производственного помещения.

Таблица 14. Оптимальные нормы параметров микроклимата

Период года	Параметр микроклимата	Величина
Холодный	Температура воздуха в помещении	22...24 °C
	Относительная влажность	40...60 %
	Скорость движения воздуха	до 0,1 м/с
Теплый	Температура воздуха в помещении	23...25 °C
	Относительная влажность	40...60 %
	Скорость движения воздуха	0,1...0,2 м/с

7.1.3. Шум

Шум является одним из наиболее распространённых в производстве вредных факторов. Действие шума не ограничивается воздействием на органы слуха, шум через нервные волокна передается в центральную и вегетативную нервные системы, а через них воздействует на внутренние органы.

Эквивалентный уровень звукового давления в помещениях для персонала, осуществляющего эксплуатацию ЭВМ, должна быть равна 50 дБА. При уровне выше 120 дБ начинаются недопустимые условия. [18] Сильный шум действует отрицательно не только на органы слуха, но и на весь организм в целом, в том числе и на нервную систему. Шум приводит к усилению утомляемости и резкому падению производительности труда.

Основным источником шума на рабочем месте оператора ПЭВМ являются вентиляторы охлаждения, трансформаторы ПЭВМ и принтер.

Уровень шума от вентиляторов и трансформаторов не превышает 45 дБ (данные взяты из технического паспорта), уровень звуковой мощности

принтера (в зависимости от модели принтера) составляет до 50 дБ, но он работает не постоянно.

Следовательно, уровень шума на рабочем месте оператора ПЭВМ следует считать допустимым.

7.1.4. Электромагнитные излучения

Электрические сигналы цепей при работе компьютера являются источниками электромагнитного излучения.

Воздействие электромагнитных полей на человека зависит от напряженностей электрического и магнитного полей, потока энергии, частоты колебаний, размера облучаемой поверхности тела и индивидуальных особенностей организма.

Нарушения в организме человека при воздействии электромагнитных полей незначительных напряжений носят обратимый характер. При воздействии полей, имеющих напряженность выше предельно допустимого уровня, развиваются нарушения со стороны нервной, сердечно–сосудистой систем, органов пищеварения и некоторых биологических показателей крови.

Наиболее мощным источником электромагнитных полей в помещении являются катушки отклоняющих систем, находящихся около цокольных частей электронно-лучевых трубок мониторов.

Согласно СанПиН 2.4.1340-03 напряженность электромагнитного поля на расстоянии 50 см вокруг ВДТ по электрической составляющей должна быть не более:

в диапазоне частот 5 Гц – 2 кГц – 25 в/м;

в диапазоне частот 2 – 400 кГц – 2.5 в/м.

Работа проводилась на современном компьютере, монитор которого отвечает стандарту ТСО'95, разработанному в Швеции, который полностью удовлетворяет требованиям по напряженности электромагнитного поля и другим показателям.

7.1.5. Освещение

Правильно спроектированное и выполненное освещение обеспечивает высокий уровень работоспособности, оказывает положительное психологическое воздействие на сотрудников, способствует повышению производительности труда.

К системам освещения предъявляют следующие требования:

- соответствие уровня освещенности рабочих мест характеру выполняемой зрительной работы;
- достаточно равномерное распределение яркости на рабочих поверхностях и в окружающем пространстве;
- отсутствие резких теней, прямой и отраженной блескости;
- постоянство освещенности во времени;
- оптимальная направленность излучаемого осветительными приборами светового потока;
- долговечность, экономичность, электробезопасность и пожаробезопасность, эстетичность, удобство и простота эксплуатации.

Согласно СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий» для обеспечения освещенности рабочего места используется естественное уличное освещение и только при его нехватке подключаются внутренние системы освещения.

Согласно действующим Санитарным правилам и нормам 2.2.2/2.4.1340-03, искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 — 500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк.

Следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПЭВМ, при этом соотношение яркости между рабочими поверхностями не должно превышать 3:1 — 5:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1.

В дисплейных залах обычно, применяют одностороннее боковое естественное освещение. С целью уменьшения солнечной инсоляции светопроемы устраивают с северной, северо-восточной или северо-западной ориентацией. Мониторы располагают подальше от окон и таким образом, чтобы окна находились сбоку.

Если экран дисплея расположен к окну, необходимы специальные экранящие устройства (светорассеивающие шторы, регулируемые жалюзи, солнцезащитная пленка с металлизированным покрытием).

О важности вопросов производственного освещения говорит и тот факт, что условия деятельности операторов в системе "человек — машина" связаны с явным преобладанием зрительной информации — до 90% общего объема.

7.2. Анализ опасных факторов

Для идентификации опасных и вредных производственных факторов определим работы и операции, связанные с выполнением ВКР. Основная работа велась на рабочем месте с использованием ПК и периферийного оборудования. Работа на ПЭВМ относится к категории работ, связанных с опасными и вредными условиями труда. Выделим опасные и вредные производственные факторы, в соответствии с и приведенными работами. [15]

Для приведенных работ вредными являются следующие факторы:

- шум на рабочем месте от работы оборудования;
- электромагнитные излучения от мониторов;
- опасность поражения электрическим током.

7.2.1. Пожарная безопасность

Пожары приводят к полной потере информации и большим трудностям восстановления всей информации в полном объёме.

Источниками воспламенения могут быть: неисправности электропроводки, розеток и выключателей которые могут привести к короткому замыканию или пробое изоляции; использование поврежденных (неисправных) электроприборов; использование в помещении электронагревательных приборов с открытыми нагревательными элементами; возникновение пожара, вследствие попадания молнии в здание; неаккуратное обращение с огнем и несоблюдение мер пожарной безопасности.

7.2.2. Электробезопасность

Электробезопасность – это система организационных и технических мероприятий и средств, обеспечивающих защиту людей от вредного и опасного воздействия электрического тока, электрической дуги и статического электричества

Электрические установки, к которым относится практически все оборудование ЭВМ, представляют для человека большую потенциальную опасность, так как в процессе эксплуатации или проведения профилактических работ человек может коснуться частей, находящихся под напряжением.

К числу опасных и вредных производственных факторов относятся повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека, повышенный уровень статического электричества, электромагнитных излучений, повышенную напряженность электрического и магнитного полей.

В отличие от других источников опасности электрический ток нельзя обнаружить без специального оборудования и приборов, поэтому воздействие его на человека чаще всего неожиданно.

Следствия воздействия электрического тока на тело человека приведены в таблице 15.

Таблица 15. Воздействие электрического тока на тело человека

Вид воздействия	Следствие	Виды электротравм
Термическое	Ожоги отдельных участков тела, нагрев внутренних органов	Электрический ожог, электрический знак, металлизация кожи.
Биологическое	Разложение и возбуждение живых тканей, судорожное сокращение мышц	Механические повреждения
Электролитическое	Разложение крови и других жидкостей, нарушение их физико-химического состава	Электрический удар

Все помещения, в зависимости от условий, делятся на помещения:

- особо опасные;
- с повышенной опасностью поражения электрическим током;
- без повышенной опасности поражения электрическим током.

В помещении, где выполнялась работа, используются приборы, потребляющие напряжение 220 В переменного тока с частотой 50 Гц. Это напряжение опасно для жизни.

Согласно классификации помещений по опасности поражения электрическим током, разработка ВКР ведется в помещении без повышенной опасности: то есть оно не должно характеризоваться наличием таких условий, как повышенная влажность (относительная влажность воздуха, длительно превышающая 75%); высокая температура (более 35 °С); токопроводящая пыль; возможность одновременного соприкосновения к имеющим соединение с землей металлическим элементам и металлическим корпусам электрооборудования.

7.3.Рекомендации по минимизации влияния

7.3.1. Рекомендации по улучшению микроклимата

К мероприятиям по оздоровлению воздушной среды в производственном помещении относятся правильная организация вентиляции и кондиционирования воздуха, отопление помещений. Вентиляция может

осуществляться естественным и механическим путём. В зимнее время в помещении необходимо предусмотреть систему отопления.

Объем помещений с ЭВМ не должен быть меньше 20 м³/человека [15].

Для обеспечения комфортных условий используются как организационные методы (рациональная организация проведения работ в зависимости от времени года и суток, чередование труда и отдыха), так и технические средства (вентиляция, кондиционирование воздуха, отопительная система).

7.3.2. Рекомендации по минимизации влияния шума

Для снижения шума следует:

- ослабить шум самих источников, используя звукоизоляцию;
- снизить эффект суммарного воздействия отраженных звуковых волн;
- использовать архитектурные и технологические решения, направленные на изоляцию источников шума.

7.3.3. Рекомендации по минимизации влияния электромагнитных излучений

Данная разработка направлена на уменьшение количества часов, затрачиваемых сотрудником на формирование статистической отчетности, что сокращает влияние электромагнитного излучения. Время на переобучение сотрудников не сильно увеличит влияние фактора, так как это происходит одноразово.

7.3.4. Рекомендации по минимизации влияния освещения

Для обеспечения требуемого уровня освещения в офисном помещении используются лампы дневного освещения, равномерно распределенные по всему потолку офиса. Для освещения помещения выбраны наиболее широко применяемые лампы типа ЛБ.

7.3.5. Меры по обеспечению пожарной безопасности

Одно из условий обеспечения пожаробезопасности любого производственного процесса – ликвидация возможных источников

воспламенения. [16] Для устранения возможности пожара в помещении должны соблюдаться следующие противопожарные меры:

- ограничение количества горючих веществ;
- устранение возможных источников возгорания (электрических искр, нагрева оболочек оборудования);
- применение средств пожаротушения;
- использование пожарной сигнализации;
- содержание электрооборудования в исправном состоянии, использование плавких предохранителей и автоматических выключателей в аппаратуре, по окончании работ все установки должны обесточиваться;
- наличие в помещении средств пожаротушения (огнетушители типа ОУ-3, пожарный инструмент, песок) и содержание их в исправном состоянии;
- разрешение курения в только отведенных для этого местах;
- содержание путей и проходов эвакуации людей в свободном состоянии;
- проводить раз в год инструктаж по пожарной безопасности;
- назначение ответственного за пожарную безопасность помещения.

7.3.6. Рекомендации по защите от электрического тока

Опасное и вредное воздействие на людей электрического тока, электрической дуги и электромагнитных полей проявляется в виде электротравм и профессиональных заболеваний.

Кроме того, при неисправности каких-либо блоков компьютера корпус может оказаться под током, что может привести к электрическим травмам или электрическим ударам. Для устранения этого предлагается обеспечить подсоединение металлических корпусов оборудования к заземляющей жиле.

При возникновении несчастного случая следует немедленно освободить пострадавшего от действия электрического тока и, вызвав врача, оказать ему необходимую помощь.

Для защиты от поражения электрическим током все токоведущие части должны быть защищены от случайных прикосновений кожухами, корпус

устройства должен быть заземлен. Заземление выполняется изолированным медным проводом сечением 1.5 мм^2 , который присоединяется к общей шине заземления с общим сечением $5,4 \text{ мм}^2$ при помощи сварки. Общая шина присоединяется к заземлению, сопротивление которого не должно превышать 4 Ом. Питание устройства должно осуществляться от силового щита через автоматический предохранитель, который срабатывает при коротком замыкании нагрузки.

При эксплуатации ЭВМ запрещается:

- включать ЭВМ при неисправной защите электропитания;
- подключать и отключать разъемы кабелей электропитания и блоков вентиляции при поданном напряжении электросети;
- заменять съемные элементы под напряжением;
- производить пайку аппаратуры, находящейся под напряжением;
- снимать щиты, закрывающие доступ к токоведущим частям;
- пользоваться электроинструментами с напряжением 36В и выше с незаземленными корпусами.

При правильной эксплуатации электроустановок и использовании соответствующих средств защиты риск поражения электрическим током сводится к минимуму.[17]

Для предотвращения поражения электрическим током в организации должны проводиться следующие мероприятия:

- компьютеры подключаются к сети с помощью трехполюсных вилок, причем центральный контакт вилки надежно заземляется.
- при эксплуатации электрооборудования рабочее место должно быть оборудовано так, что исключается возможность прикосновения служащих к токоведущим устройствам, шинам заземления, батареям отопления, водопроводным трубам.
- обслуживающий персонал должен пройти инструктаж по технике безопасности на рабочем месте.

- осуществляется профилактическая проверка отсутствия напряжения, отключение неисправного оборудования и наложение заземления.

7.4 Экологическая безопасность

7.4.1. Анализ воздействия на окружающую среду

Охрана окружающей среды - деятельность органов государственной власти Российской Федерации, органов государственной власти субъектов Российской Федерации, органов местного самоуправления, общественных объединений и некоммерческих организаций, юридических и физических лиц, направленная на сохранение и восстановление природной среды, рациональное использование и воспроизводство природных ресурсов, предотвращение негативного воздействия хозяйственной и иной деятельности на окружающую среду и ликвидацию ее последствий. Наиболее активной формой защиты окружающей среды от вредного воздействия выбросов промышленных предприятий является полный переход к безотходным, малоотходным и энергосберегающим технологиям и производствам.

При разработке любых автоматизированных систем возникает необходимость утилизировать производственные отходы, в качестве которых в данном случае выступают бумажные отходы (макулатура) и неисправные детали персональных компьютеров, плат, контроллеров.

7.4.2. Рекомендации по минимизации влияния на окружающую среду

Бумажные отходы должны передаваться в соответствующие организации для дальнейшей переработки во вторичные бумажные изделия. Неисправные комплектующие персональных компьютеров должны передаваться либо государственным организациям, осуществляющим вывоз и уничтожение бытовых и производственных отходов, либо организациям, занимающимся переработкой отходов. Важнейшим этапам обращения с отходами является их сбор, а в дальнейшем переработка, утилизация и захоронение[18].

Еще одним из способов снижения бумажных отходов является хранение данных на электронных носителях. Данное ПО позволяет хранить и

предоставлять статистические данные на административных серверах компании ООО «Томскнефтехим», в электронном виде.

7.5 Безопасность в чрезвычайных ситуациях

7.5.1 Перечень возможных чрезвычайных ситуаций на объекте

При работе в компании ООО «ИТ-СЕРВИС» могут возникнуть следующие классификации чрезвычайных ситуаций:

- преднамеренные/непреднамеренные;
- техногенные: взрывы, пожары, обрушение помещений, аварии на системах жизнеобеспечения/природные – связанные с проявлением стихийных сил природы.
- экологические – это аномальные изменения состояния природной среды, такие как загрязнения биосферы, разрушение озонового слоя, кислотные дожди/ антропогенные – являются следствием ошибочных действий людей.
- биологические – различные эпидемии, эпизоотии, эпифитотии;
- комбинированные.

По скорости развития чрезвычайные ситуации могут быть: взрывные, внезапные, скоротечные, плавные. По масштабам распространения последствий: локальные, местные, территориальные. По возможности предотвращения: неизбежные (например, природные) и предотвращаемые (например, техногенные, социальные).

Одной из наиболее возможных ЧС в офисном здании является возникновение пожара. Причинами возникновения данного вида ЧС могут являться:

- возникновением короткого замыкания в электропроводке;
- возгоранием устройств ПЭВМ из-за неисправности аппаратуры;
- возгоранием устройств искусственного освещения;
- возгоранием мебели по причине нарушения правил пожарной безопасности, а также неправильного использования дополнительных бытовых электроприборов и электроустановок.

В данной разработке для хранения конфиденциальных данных используются электронные БД, которые позволяют сотрудникам не беспокоиться о том, что их необходимо эвакуировать, в отличие от бумажных носителей. Электронные БД не хранятся на компьютере в обычном помещении учреждения, для этих целей используются серверы, которые стоят в серверной комнате. Серверы снабжаются датчиками, которые реагируют на окружающую среду и при необходимости могут создавать бэкапы данных и отправлять данные на резервный сервер или в облачное хранилище.

7.5.2 Меры по ликвидации чрезвычайных ситуаций и последствий

Пожарная безопасность подразумевает надлежащее состояние объекта с исключением возможности возникновения очага возгорания (пожара) и его распространения в пространстве. Обеспечение пожарной безопасности — приоритетная задача для любого предприятия. Создание системы защиты регламентировано законом и нормативными документами различных ведомств. Противопожарный режим для каждого предприятия является индивидуальным. Он должен отвечать общим требованиям, установленным законодательными актами, но с учетом специфики деятельности предприятия.

В распорядительном документе на предприятии ООО «Томскнефтехим» отображаться следующая информация:

1. Определение и создание мест для курения;
2. Четкое установление места и количества находящегося в помещении сырья, готовой продукции и т. п.;
3. Составление графика уборки;
4. Установление режима хранения спецодежды;
5. Порядок отключения источника энергоснабжения в штатной ситуации и в случае возникновения ЧС.

Правила хранения конфиденциальных данных в электронных таблицах. В киберпространстве могут быть использованы различные способы для совершения кибертеракта:

- получение несанкционированного доступа к государственным и военным секретам, банковской и личной информации;
- нанесение ущерба отдельным физическим элементам информационного пространства, например, разрушение сетей электропитания, создание помех,
- использование специальных программ для разрушения аппаратных средств;
- кража или уничтожение информации, программ и технических ресурсов путем преодоления систем защиты, внедрения вирусов, программных закладок;
- воздействие на программное обеспечение и информацию;
- раскрытие и угроза публикации закрытой информации;
- захват каналов СМИ с целью распространения дезинформации, слухов, демонстрации мощи террористической организации и объявления своих требований;
- уничтожение или активное подавление линий связи, неправильная адресация, перегрузка узлов коммуникации;
- проведение информационно-психологических операций.

Направления для защиты от кибератак :

- мониторинг и реагирование на инциденты (включая поведенческий анализ пользователей);
- анализ исходных кодов ПО;
- сетевая безопасность;
- шифрование.

Основными причинами возникновения ЧС являются: сложность технологий, недостаточная квалификация персонала, проектно-конструкторские недоработки, низкая технологическая дисциплина, а также внешние чрезвычайные ситуации: это стихийные бедствия.

7.6.Правовые и организационные вопросы обеспечения безопасности

7.6.1 Психофизиологические факторы

К психофизиологическим вредным факторам относятся статические физические перегрузки, умственное перенапряжение, монотонность труда, эмоциональные перегрузки.

Организация работы с ПЭВМ должна осуществляться в зависимости от вида и категории трудовой деятельности. Для предупреждения преждевременной утомляемости пользователей ПЭВМ рекомендуется организовывать рабочую смену путем чередования работ с использованием ПЭВМ и без него. Во время регламентированных перерывов с целью снижения нервно-эмоционального напряжения, утомления зрительного анализатора, устранения влияния гиподинамии и гипокинезии, предотвращения утомления целесообразно выполнять комплексы упражнений.

7.6.2. Организационные мероприятия при компоновке рабочей зоны

Большое значение для профилактики статических физических перегрузок имеет правильная организация рабочего места человека, работающего с ПЭВМ. Рабочее место должно быть организовано в соответствии с требованиями стандартов, технических условий и (или) методических указаний по безопасности труда. Конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации и т.д.) должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы.

Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5 - 2,0 м. Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600 - 700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления.

Так же стоит учитывать требования к размещению средств отображения информации (Таблица 16).

Таблица 16 – Расположение средств отображения информации

Тип средств отображения информации	Угол, градусы	
	В вертикальной плоскости	В горизонтальной плоскости
Очень часто используемые	± 15	± 15
Часто используемые	± 30	± 30
Редко используемые	± 60	± 60

7.6.3. Обеспечение гарантий защиты конфиденциальных данных граждан

При заключении договора, подписывается «Соглашение о конфиденциальности и неразглашении информации». Его целью является урегулирование вопроса хранения, использования, разглашения и передачи конфиденциальной информации.

В документе «Соглашение о конфиденциальности» можно выделить следующие разделы:

- Предмет соглашения. Содержит цель и предназначение документа.
- Права и обязанности сторон.
- Ответственность сторон. В данном пункте указаны факторы, при которых раскрывающая сторона несет юридическую ответственность за разглашение, использование либо нарушение условий хранения конфиденциальной информации. Учитываются пункты как умышленного, так и непреднамеренного пользования конфиденциальной информацией. А также ряд юридических санкций, в случае нарушения вышеуказанных положений.

- Условия раскрытия информации. Указываются строгие критерии и условия, при которых данная информация может быть раскрыта. Первым условием обязано стоять «обоюдное согласие сторон». Без данного фактора информация к разглашению должна быть запрещена.

Данный документ составлен на основе Федеральных законов[26-27].

Также предусмотрен комплекс технических мер предупреждения утечки информации, который приведен ниже.

Защита баз СУБД от действий системных администраторов. Для хранения используется Oracle 11g, который, защищает БД от действий системного администратора, благодаря трём новым разрешениям уровня сервера (CONNECT ANY DATABASE, IMPERSONATE ANY LOGIN, и SELECT ALL USER SECURABLES), появилась возможность более гибко управлять серверными ролями. В том числе есть возможность можем создать администраторскую серверную роль с полным запретом на доступ к пользовательским данным.

Шифрование конфиденциальной информации. Обеспечивает дополнительный уровень безопасности конфиденциальных данных, защищая информацию от посторонних пользователей. При обращении к сервису авторизации пользователь должен ввести свой логин и пароль, затем ему приходит сообщение на e-mail или по sms. Таким образом, реализуется двухфакторная аутентификация администратора при входе в систему.

Независимое резервное копирование. Каждое утро на сервере Заказчика создается резервная копия БД, что дает уверенность в актуальной копии всех важных данных прямо у него в руках.

Заключение

В любой организации, как большой, так и маленькой, возникает проблема такой организации управления данными, которая обеспечила бы наиболее эффективную работу. Небольшие организации используют для этого шкафы с папками, однако крупные корпоративные предприятия используют компьютеризированные системы автоматизации, позволяющие эффективно хранить, извлекать информацию и управлять большими объемами данных.

Темпы внедрения новых технологий в компьютерной отрасли вызывают изумление. Компании, конкурирующие за рынки и прибыли, стремятся моментально реализовать технические новшества в аппаратных средствах, программном обеспечении и парадигмах вычислений, стимулирующих развитие всей технологии управления информацией. Однако для успешной реализации крупных систем управления требуется применить нестандартный подход, творческое решение. Использование основ эргономики при проектировании, реализации и внедрении системы управления позволит решить многие «психологические» и «технологические» проблемы предприятий.

В процессе разработки программы основная проблема состояла в организации дружественного пользовательского интерфейса и организации стабильной работы всех функций программы, связанных как непосредственно с функциями управления данными, хранящимися в базе данных, так и связанных с сервисными функциями программы.

Программа предназначена для автоматизации процесса регистрации обращений пользователей, а так же для обработки этой информации.

В процессе проектирования были выявлены наиболее существенные характеристики объекта, разработана концептуальная модель предметной области.

Значительное внимание уделено детальной разработке архитектуры информационной системы в целом, а так же проектных решений по отдельным ее объектам и элементам, их анализу, и внедрению. На этапе рабочего проектирования созданы таблицы базы данных и приложения, реализующие все

поставленные задачи и обеспечивающие пользователю удобную работу с информацией.

При разработке данного программного продукта была использована среда разработки Delphi 2009 компании CodeGear. Приложение, разработанное в данном дипломном проекте, полностью совместимо с операционной системой Windows. Компоненты, применяемые в разработке, имеют аналогичный вид и наделены теми же свойствами, что и стандартные компоненты Windows.

Список публикаций

1. Аксёнов С.В., Быков А.В. Алгоритм качества оценки кластеризации. Молодёжь и современные информационные технологии: сборник трудов XII Международной научно-практической конференции студентов, аспирантов и молодых ученых, г. Томск, 12-14 ноября 2014 г. Национальный исследовательский Томский политехнический университет, 2014 – С. 90-91.

2. Аксёнов С.В., Быков А.В. Исследование современных приложений ИТ-поддержки. Разработка базы данных предварительной диспетчеризации обращений пользователей. Информационные технологии в науке, управлении, социальной сфере и медицине. III Международная научная конференция, 23-26 мая 2016 Национальный Исследовательский Томский Политехнический университет.

Список использованных источников

1. Мишель Е. Дэвис. Изучаем PHP и MySQL / Мишель Е. Дэвис, Джон А. Филлипс.; - М.:Изд-во «Символ-Плюс» - 422с,2008г.
2. Различные архитектурные решения, используемые при реализации многопользовательских СУБД [Электронный ресурс] / ИНТУИТ Национальный открытый университет. URL: <http://www.intuit.ru/> , свободный. – Загл. с экрана. – Яз.рус.,англ. Дата обращения 25.05.2016
3. Прохоров А.А. «Проектирование баз данных и информационных систем» [Электронный ресурс]/Прохоров А.А.. «Пермская ГСХА имени академика Д.Н. Прянишникова» URL: <http://any-book.org/download/2152.html>, свободный. – Загл. с экрана. – Яз.рус.,англ. Дата обращения 25.05.2016
4. Сибилев В.Д. Модели и проектирование баз данных Часть 2, Томск, ТМЦДО -144 с, 2002 г.
5. Фараонов В.В., Delphi 2009, учебный курс, 1-ое издание, 2010г. – 672с.
6. Компьютерные технологии в высшем образовании / А.Н. Тихонов [и др.]. - М.: Изд-во Моск. Ун-та.-370 с
7. Архангельский А.Я., Программирование в Delphi. — М.: ООО «Бином-Пресс», 2009 г. — 1152 с
8. Диго С.М. Учебное пособие по курсу “Базы данных” / Московский международный институт эконометрики, информатики, финансов и права. - М., 2008. – 177 с
9. Справочник – «Основы Delphi» [Электронный ресурс] / Под ред. П.В. Шумакова . URL: <http://www.delphibasics.ru/>, свободный Загл. с экрана. – Яз.рус.,англ. Дата обращения 12.12.2015
10. Основы современных баз данных [Электронный ресурс] / Под ред. С. Кузнецова - Центр Информационных Технологий. URL: <http://www.citforum.ru/database/osbd/contents.shtml>, свободный Загл. с экрана. – Яз.рус.,англ. Дата обращения 18.12.2015

11. Основы проектирования реляционных баз данных [Электронный ресурс] / Под ред. В.В. Кириллов - Санкт-Петербургский Государственный институт точной механики и оптики. URL:<http://www.citforum.ru/database/dbguide/index.shtml>, свободный Загл. с экрана. – Яз.рус.,англ. Дата обращения 28.08.2015
12. Системные требования Oracle [Электронный ресурс]/ ELMA|База знаний. URL: <http://www.elma-bpm.ru/kb/article-259.html>, свободный Загл. с экрана. – Яз.рус.,англ. Дата обращения 28.11.2015
13. Справочник трудового кодекса [Электронный ресурс]. URL:<http://tomsk.hh.ru/>,свободный Загл. с экрана. – Яз.рус.,англ. Дата обращения 08.01.2016
14. Федеральный закон от 24.07.2009 N 212-ФЗ « О страховых взносах в Пенсионный фонд Российской Федерации, Фонд социального страхования Российской Федерации, Федеральный фонд, обязательного медицинского страхования»
15. ГОСТ 12.0.003-74. ССБТ Опасные и вредные производственные факторы. Классификация. – М.: ИПК Издательство стандартов, 2002.
16. ГОСТ 12.0.002-80. Система стандартов безопасности труда. Термины и определения. – М.: ИПК Издательство стандартов, 2002.
17. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. – М.: Информационно-издательский центр Минздрав России, 2003.
18. СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений. – М.: Информационно-издательский центр Минздрав России, 1997.
19. ГОСТ 12.1.005-88. Общие санитарно-гигиенические требования к воздуху рабочей зоны. – М.: Издательство стандартов, 1998.
20. СанПиН 2.2.4.1294-03. Гигиенические требования к аэроионному составу воздуха производственных и общественных помещений. – М.: Информационно-издательский центр Минздрав России, 2003.

21. СНиП 23-05-95. Естественное и искусственное освещение. – М.: Минстрой России, 1995.
22. СНиП 2.2.4/2.1.8.562-96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки. – М.: Информационно-издательский центр Минздрав России, 1996.
23. СН 2.2.4/2.1.8.566-96. Производственная вибрация, вибрация в помещениях жилых и общественных зданий. – М.: Минстрой России, 1997.
24. ГОСТ 22.0.07-95. Безопасность в чрезвычайных ситуациях. Источники техногенных чрезвычайных ситуаций. Классификация и номенклатура поражающих факторов и их параметров. – М.: Госстандарт России, 1995.
25. ГОСТ Р 55090-2012 Ресурсосбережение. Обращение с отходами. Рекомендации по утилизации отходов бумаги
26. Федеральный закон от 29.07.2004 №98-ФЗ (ред. от 11.07.2011) «О коммерческой тайне». Федеральный закон «О защите персональных данных» ФЗ-152.
27. Федеральный закон «Об информации, информационных технологиях и о защите информации».
28. Бочаров В. В. Инвестиции: учебник для вузов / В. В. Бочаров. – 2-е изд. – СПб.: Питер, 2009. – 381 с. – Учебник для вузов.
29. Староверова Г. С. Экономическая оценка инвестиций : учебное пособие / Г. С. Староверова, А. Ю. Медведев, И. В. Сорокина. – 2-е изд., стер. – Москва: КноРус, 2009. — 312 с
30. Голосовский С. И. Эффективность научных исследований в промышленности / С. И. Голосовский. – Москва: Экономика, 1986. – 159 с.
31. Мигуренко Р. А. Научно-исследовательская работа: учебно-методическое пособие / Р. А. Мигуренко; Национальный исследовательский Томский политехнический университет (ТПУ), Институт дистанционного образования (ИДО). – 2-е изд., стер. – Томск: Изд-во ТПУ, 2010. – 184 с.

Приложение А

Раздел 4 *Circuit Implementation Procedure*

Студент:

Группа	ФИО	Подпись	Дата
8BM4B	Быков Александр Вадимович		

Консультант кафедры оптимизации систем управления:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Чердынцев Е.С.	к.т.н.		

Консультант – лингвист кафедры иностранных языков института кибернетики:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Куркан Н.В.			

4. Circuit Implementation Procedure

4.1. Choosing a network architecture type

Three-tiered architecture represents a further improvement of the client-server technology. All business logic that was previously included into the client applications is separated in a disjoint component called the application server in the three-tiered architecture. At the same time, client applications run on the user system interface client environment.

As a result, the process is arranged as follows:

- A database in the form of a file set is stored on a hard drive of a dedicated PC (network drive);
- DBMS is located on a network server;
- There is a dedicated application server that includes business analysis software;
- there is a number of client PCs provided with a Thin Client (a special application that implements a user interface);
- users may run the Thin Client on each client PC. With the help of the user interface, provided by the application, it initiates the procedure reference to the business analysis software, located at the application server;
- Application server analyzes user requirement and builds up queries to the DB. A special query language SQL is used for communication, i.e. only the query text is streamed through the network from the application server to the DB server;
- the DB encapsulates all physical structure data of the server DB;
- the DB initiates queries to the server data that results in replication of the query outcome to the application server;
- the application server returns the result to the client application (to the user);
- by the aid of the user interface the application displays query outcomes.

The three-tiered client-server architecture represented in Figure 14 has been used in this thesis.

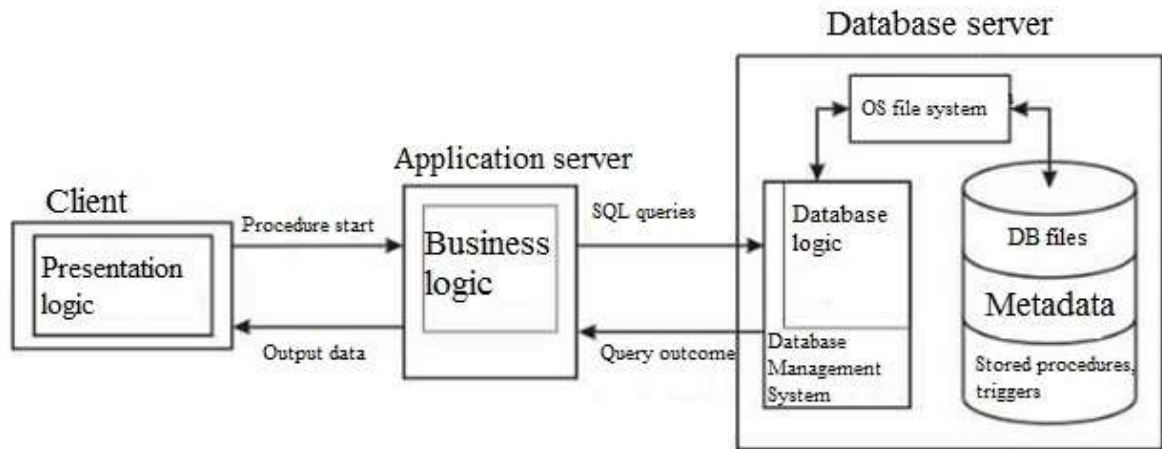


Figure 14. Three-tiered architecture

The purpose of this thesis is to develop a user request preliminary dispatch database by the example of IT SERVICE Ltd.

4.2. Description of DB constraints

DB constraints are set to invoke arrangement regulations on the tabulated data. When the constraint is on, all tabulated data are subject to the regulations attributed. If after that there runs SQL statement that alters data in this table, then ORCALE validates the outputs against their compliance with the constraints.

The existing reference constraints in the operated DB are described below:

- in case POSITION entry is deleted, provided that there are related entries in table INITIATOR, internal ORCALE exception is activated and the task becomes undone;
- in case COMPANY entry is deleted, provided there are related entries in table DEPARTMENT, internal ORCALE exception is activated and the task becomes undone;
- in case DEPARTMENT entry is deleted, provided there are related entries in table DIVISION, internal ORCALE exception is activated and the task becomes undone;

- in case DIVISION entry is deleted, provided there are related entries in table INITIATOR, internal ORCALE exception is activated and the task becomes undone;
- in case INITIATOR entry is deleted, provided there are related entries in table REQUEST, internal ORCALE exception is activated and the task becomes undone;
- in case AGREEMENT entry is deleted, provided there are related entries in table INITIATOR, internal AGREEMENT SERVICES exception is activated and the task becomes undone;
- in case SERVICE entry is deleted, provided there are related entries in table AGREEMENT SERVICES, internal ORCALE exception is activated and the task becomes undone;
- in case AGENT entry is deleted, provided there are related entries in table REQUEST, internal ORCALE exception is activated and the task becomes undone;
- in case AGREEMENT SERVICES entry is deleted, provided there are related entries in table REQUEST, internal ORCALE exception is activated and the task becomes undone;

The existing data integrity constraints are described in the form of Table 3 in Appendix B.

4.3. Database administration

Administration of any database management system is restricted to the following number of tasks:

1. Protection against unauthorized access;
2. User administration and accessing clients to the data;
3. Database archiving and recovery, backup schedule arrangement.

4.3.1. Protection against unauthorized access

Every PC in the network requires constant protection against malicious software, such as worms, spyware, remote controllers that may cause harm and malfunction from slow-up to identity theft.

Firewall is most often used against unauthorized access.

Firewall is a software or hardware complex for access control to local or global networks. Its basic principle involves tracing of all established connections and further browsing of the list of rules (set by the user or administrator) with final resolution to permit or deny the connection.

Firewall may be divided into corporate and personal.

There are two types of corporate Firewall: software and hardware. They are installed on a gateway between the local network and Internet (or on a gateway between two sub-networks) and use protocol type, web-page address and protocol number as verification rules. Corporate Firewall rules are established by the network administrator.

Personal Firewall is software installed directly on a user's PC. The principal difference of personal Firewall from the corporate one is that in personal Firewall one may set not only protocol, address and port for connection, but also software with a permission to establish the connection (or rather to accept inbound connection). Almost all personal Firewall systems have a training mode, i.e. if there is no rule for any software, the user is notified that the software made an attempt to establish connection at the referred address and then the user should decide whether to allow or deny the connection. Accordingly, if you choose ALLOW, then the software will be able to establish connection and send data. Thus, Firewall is not a cross-functional solution of all security problems, as it may seem. Firewall is a tool that helps to protect computer against attacks, but only if you know what you do.

One should follow the instructions below for proper use of Firewall:

1. read Firewall notifications attentively;
2. allow access only to the required software;
3. allow access only to the required ports used by protocols and a software program;
4. in uncertain cases, please, follow the principle – spare “deny” is better than spare “allow”.

Table 17 serves as a reference to determine the type of ports compatible with various protocols.

In most cases no incoming Internet connections, except connections from local address 127.0.0.1, should be accepted at common user PC. This IP-address is a computer home address (connections to this address may be established by various filter programs, e.g. Proxomitron, or local proxy servers like CoolProxy) and one may deny all requests on incoming connection acceptance (except for the cases, when the purpose of the incoming connection is evident). There may be following exceptions: FTP-clients may accept incoming connections in the active mode, mIRC accepts incoming connection to port 113 when connecting to the server and in some certain situations ICQ may accept incoming connections.

Now, there is a great number of personal Firewall systems: Augnitum Outpost, ZoneAlarm, Kerio Personal Firewall, Kaspersky Antihacker, Ms Trend Micro, Tiny Personal Firewall, etc. Almost all of them have a similar operating principle and differ in interface and usability. Oracle uses integrated Oracle DataBase Firewall.

Table 17 – Correlation of ports and protocols

Program type	Protocol	Port
DNS-client-determination of address by symbolic name (SVCHOST.EXE)	DNS	53 (outbound TCP and UDP)
Browser (Internet Explorer, Opera, Mozilla)	HTTP, HTTPS	80, 443 (outbound TCP)
E-mail client (The Bat, Outlook, Eudora, Becky) – outbox.	SMTP	25 (outbound TCP)
E-mail client (The Bat, Outlook, Eudora, Becky) – inbox.	POP3, IMAP	110 (outbound TCP)
FTP-client (Windows Commander, CuteFTP, SmartFTP, BulletProof FTP)	FTP	21 (outbound TCP), ports within 2000-4000 (outbound TCP)
ICQ-client (ICQ, &RQ, Miranda, Trillian)	ICQ	5190 (outbound TCP)
IRC-client (mIRC)	IRC, Ident	6666-6669 (outbound TCP), 113 (inbound TCP)

4.3.2. User administration and accessing clients to the data

User accounts of Oracle databases holds a prominent place in security tools.

CREATE USER is the SQL statement that may be used to determine Oracle account in the database. After Oracle user account is created, it may not be in use before the user is assigned with at least one system privilege. The CREATE SESSION allows the user to create session towards Oracle database. This is an essential privilege of the user account, because Oracle user account cannot be used without it.

At initial creation of the Oracle user a tablespace set by default, where user objects will be created, can be determined. If a tablespace set by default is not determined, the user will be assigned with the SYSTEM tablespace by default, which will be used by the database objects. The DEFAULT TABLESPACE statement may be used in the CREATE USER statement to make all user objects be placed in the tablespace different from the SYSTEM one. Oracle user should also be assigned with a quota determining the memory volume it may use in the tablespace.

Another way to create a user is to provide the user with the CONNECT, RESOURCE and DBA roles. Though this is a quick method, it is primarily included for compatibility with previous Oracle software releases. The Create USER statement is a preferable method, as one statement may contain both quota and other settings.

The ALTER USER statement may be used to alter such user parameters as password, temporary tablespaces set by default and memory quota.

The DROP USER statement is used to remove a database user – it purges user account from the Oracle data dictionary.

If Oracle user has any database objects, each object may be deleted before the DROP USER statement application, or use the CASCADE option in DROP USER for automatic deletion of all objects at user account removal.

Database damages may occur during the DBMS operation due to hardware malfunction, insufficient power supply, user-defined or program errors and various man-made factors. It is therefore necessary to create database and transaction log backups. Database recovery process includes its backup recovery, introduction of

updates from the transaction log made up between the backup creation and the last log backup and, if necessary, manual recovery of updates, introduced between the last transaction log backup and data loss.

A backup may also be damaged, for this reason it is recommended to store backups on different storage devices (removable hard disks, magnetic tapes, etc.), which cannot be damaged together with hardware. Besides, is recommended to store previously used backups.

Backup and recovery process is fully automated. It is possible to make up a backup arrangement according to which the procedure will be automatically sequenced by time for the designated databases.

Oracle records all changes introduced into the memory data blocks, online redo logs and generally does that before their recording into the database files. Oracle uses all changes recorded in the log files for the database update at the recovery process. Oracle has two file control modes.

ARCHIVELOG mode

In the ARCHIVELOG mode Oracle backs up (archives) the completed redo logs. Therefore, no matter how long ago the backup was performed, the database can be always recovered to any point with the help of archive logs in the ARCHIVELOG mode.

NOARCHIVELOG mode

In this mode the completed redo logs are rewritten, not saved. This, accordingly, means that in the NOARCHIVELOG mode only backup recovery may be performed and all other changes introduced into the database after the backup will be lost. This mode ensures possibility to recover only after the database instance failure. In case of data storage troubles (e.g. disk loss), the database operating in the NOARCHIVELOG mode may be recovered only from the backup with a definite loss of all changes introduced after creating of the backup.

Almost all production databases operate in the ARCHIVELOG mode. The NOARCHIVELOG mode is used only if there is a possibility for data recovery from

other sources or if the database is at the stage of development or testing and thus does not require data recovery with a minute precision.

Both the whole database and a part of it in the form of its tablespace or data file may be subject to backup. Please, note that in case the database operates in the NOARCHIVELOG mode, it is forbidden to conduct partial database backup, provided all the tablespaces and files subject to the backup, are read-only. Whole database backup may be implemented in both ARCHIVELOG and NOARCHIVELOG modes.

Whole database backup is more frequent. It proposes backup of the important control file among all data file backup. Oracle will not open the database without control file; therefore the latest control file backup as well as all data file backups is required for the recovery.

Figure 15 represents database recovery scheme.

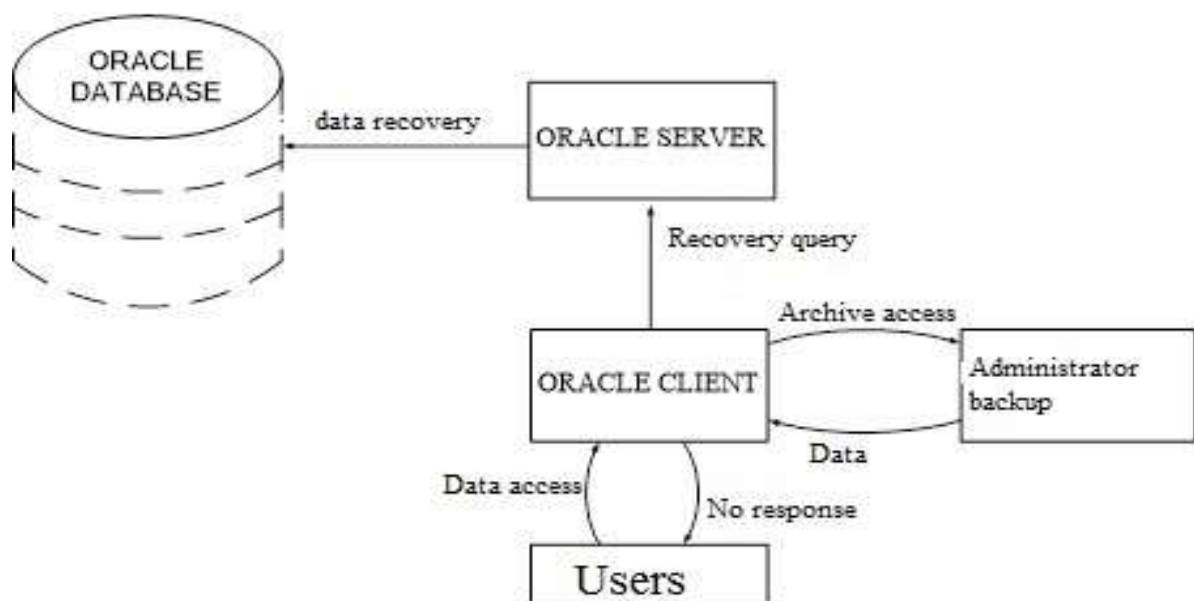


Figure 15 Data recovery scheme

4.4. Hardware requirements

The following configuration will be sufficient for proper Oracle 11g DBMS server [11]:

1. Database server, application server:

- Intel Xeon CPU 3.0 GHz;
- 16 Gb RAM;
- free HDD space – 200 Gb.

2. Workstation:

- Pentium 4 2.5 GHz;
- 2 Gb RAM;
- Free HDD space – 40 Gb
- Windows XP SP2 OS and higher
- input devices (keyboard, mouse)
- output devices (monitor display)
- local network bandwidth – from 10 Mbps (optimum – 100 Mbps)

Приложение Б

Таблица 1- Сравнительные характеристики существующих СУБД

Характеристики	Платформа			
	Oracle	FireBird	MS Access	My SQL
Соотношение цена/качество	Наиболее мощная платформа, но и самая дорогостоящая	Бесплатная и при этом довольно серьезная	Хорошие возможности и недорого	Бесплатная но и не умеет ничего, кроме выполнения простых запросов
Круг задач	Корпоративные системы	Информационные системы для малого бизнеса	Офисные базы данных и системы на их основе	Web-сайты
Возможности общие	Выполнение SQL- запросов (ANSI'92), поддержка транзакций	Выполнение SQL- запросов (ANSI'92), поддержка транзакций	Выполнение SQL- запросов (ANSI'92), поддержка транзакций	Выполнение SQL- запросов (ANSI'92), поддержка транзакций
Возможности специфические	Хранимые процедуры, функции, представления, триггеры, пакеты. Языки PL/SQL	Хранимые процедуры, функции, представления, триггеры. Свой диалект языка SQL	Представления (запросы). Свой диалект языка SQL и VBScript	Разные типы таблиц (с поддержкой транзакций и без них)
Удобство разработки и администрирования	Очень мощные, но и сложные средства администрирования	Слабые средства администрирования	Развитые средства администрирования	Собственные отсутствуют, но есть сторонние разработки

Продолжение таблицы 1

Характеристики	Платформа			
	Oracle	FireBird	MS Access	My SQL
Поддержка экспорта и импорта данных из других форматов	Экспорт и импорт в текстовые файлы	Экспорт и импорт в текстовые файлы	Экспорт и импорт в любой источник ODBC	Отсутствует
Общая оценка	90	50	60	55

Приложение В

Таблица 3- Ограничения целостности данных

ТАБЛИЦА	АТРИБУТ	NULL-ПОЛЕ	ОГРАНИЧЕНИЯ ЦЕЛОСТНОСТИ
ДОЖНОСТЬ	Код должности	NOT NULL	Целое
ДОЖНОСТЬ	Наименование	NOT NULL	
ИНИЦИАТОР	Код инициатора	NOT NULL	Целое
ИНИЦИАТОР	Табельный номер	NOT NULL	Текст, не более 15 символов
ИНИЦИАТОР	Фамилия	NOT NULL	Текст, не более 15 символов
ИНИЦИАТОР	Имя	NOT NULL	Текст, не более 15 символов
ИНИЦИАТОР	Отчество	NOT NULL	Текст, не более 15 символов
ИНИЦИАТОР	Статус		Текст, не более 20 символов
ИНИЦИАТОР	Телефон		Текст, не более 15 символов
ИНИЦИАТОР	Email		Текст, не более 30 символов
ИНИЦИАТОР	Код отдела	NOT NULL	Целое
ИНИЦИАТОР	Код должности	NOT NULL	Целое
ОРГАНИЗАЦИЯ	Код организации	NOT NULL	Целое
ОРГАНИЗАЦИЯ	Наименование	NOT NULL	Текст, не более 200 символов
ОРГАНИЗАЦИЯ	Физический адрес		Текст, не более 1000 символов
ОРГАНИЗАЦИЯ	Юридический адрес		Текст, не более 1000 символов
ОРГАНИЗАЦИЯ	ИНН		Целое, не менее 10 знаков, не более 12 знаков
ДЕПАРТАМЕНТ	Код департамента	NOT NULL	Целое
ДЕПАРТАМЕНТ	Наименование	NOT NULL	Текст, не более 200 символов
ТАБЛИЦА	АТРИБУТ	NULL-ПОЛЕ	ОГРАНИЧЕНИЯ ЦЕЛОСТНОСТИ
ДЕПАРТАМЕНТ	Код организации	NOT NULL	Целое
ОТДЕЛ	Код отдела	NOT NULL	Целое
ОТДЕЛ	Наименование	NOT NULL	
ОТДЕЛ	Код департамента	NOT NULL	Целое
ТАБЛИЦА	АТРИБУТ	NULL-ПОЛЕ	ОГРАНИЧЕНИЯ ЦЕЛОСТНОСТИ
ИСПОЛНИТЕЛЬ	Код исполнителя	NOT NULL	Целое

Продолжение таблицы 3

ИСПОЛНИТЕЛЬ	Фамилия	NOT NULL	Текст, не более 15 символов
ИСПОЛНИТЕЛЬ	Имя	NOT NULL	Текст, не более 15 символов
ИСПОЛНИТЕЛЬ	Отчество	NOT NULL	Текст, не более 15 символов
ИСПОЛНИТЕЛЬ	Телефон		Текст, не более 15 символов
ИСПОЛНИТЕЛЬ	Email		Текст, не более 30 символов
ОБРАЩЕНИЕ	Код обращения	NOT NULL	Целое
ОБРАЩЕНИЕ	Код инициатора	NOT NULL	Целое
ОБРАЩЕНИЕ	Код исполнителя	NOT NULL	Целое
ОБРАЩЕНИЕ	Дата обращения	NOT NULL	Дата
ОБРАЩЕНИЕ	Тема	NOT NULL	Текст, не более 100 символов
ОБРАЩЕНИЕ	Описание		Текст, не более 4000 символов
ОБРАЩЕНИЕ	Приоритет		Целое, не менее 1, не более 10
ОБРАЩЕНИЕ	Затрачено время		Число, не более пяти знаков после запятой
ОБРАЩЕНИЕ	Код услуги договора	NOT NULL	Целое
ДОГОВОР	Код договора	NOT NULL	Целое
ДОГОВОР	Номер договора	NOT NULL	Текст, не более 50 символов
ДОГОВОР	Дата договора	NOT NULL	Дата
ДОГОВОР	Начало действия		Дата
ДОГОВОР	Окончание действия		Дата
УСЛУГА	Код услуги	NOT NULL	Целое
УСЛУГА	Наименование услуги	NOT NULL	Текст, не более 1000 символов
УСЛУГИ ДОГОВОРА	Код услуги договора	NOT NULL	Целое
УСЛУГИ ДОГОВОРА	Код договора	NOT NULL	Целое
ТАБЛИЦА	АТТРИБУТ	NULL-ПОЛЕ	ОГРАНИЧЕНИЯ ЦЕЛОСТНОСТИ
УСЛУГИ ДОГОВОРА	Код услуги	NOT NULL	Целое

ПРИЛОЖЕНИЕ Б

Приложение Г

Таблица 5 – Сравнения аналогов с новым продуктом

Возможности программных продуктов	Программные продукты (Аналоги)		
	Mosaic	OmniTracker	Новый разрабатываемый продукт
1.Регистрация обращения	+	+	+
2.Изменение статуса обращения	-	+	+
3.Изменение инициатора	+	+	+
4.Изменение приоритета обращения	-	-	+
5.Диспетчеризация обращений на продуктовые функционирующие группы ИТ-специалистов	+	+	+
6.Контроль сроков обработки обращений (время создания обращения, время назначения, время выполнения обращения, время фактического выполнения, время закрытия обращения)	-	+	+
7.Производство диспетчеризации нарядов на продуктовые функционирующие группы ИТ-специалистов	-	-	+
8.Изменение приоритета нарядов	-	-	+
9.Контроль сроков обработки наряда (время создания наряда, время назначения, время выполнения наряда, время факт. выполнения, время закрытия наряда)	-	-	+
10.Создание проблемы	-	-	+

Продолжение таблицы 5

11.Связывания сущности обращения с сущностями нарядов/изменений/проблемы	-	-	+
12.Синхронизация информации базы данных с Active Directory	-	+	+
13. Добавление статуса и изменения шрифта обращения в статусе «Просрочено»	-	+	+
14.Экспортизация необходимой информации	-	+	+
15.Создание/Настройка доп. представлений необходимых пользователю	-	+	+
16.Автоматическая проверка на орфографию поля «Решение»	-	-	+
17.Синхронизация информации базы данных с КХД	-	-	+
18.Интеграция системы со смежными системами 1С ИТIL/SSM	-	-	+/- (на доработке)
19.Автоматическая регистрация обращений	-	-	+
20.Проставление оценок инициатором по обращению	-	-	+
21. Повторный возврат обращения в работу (реджект)	+	-	+

Приложение Д

Листинг генерации схемы БД

```
-- Create the user
  create user NProject
  identified by NProject
  default tablespace SYSTEM
  temporary tablespace TEMP
  profile DEFAULT;
-- Grant/Revoke role privileges
grant connect to NProject;
grant resource to NProject;
-- Grant/Revoke system privileges
grant create materialized view to NProject;
grant create procedure to NProject;
grant create public synonym to NProject;
grant create role to NProject;
grant create synonym to NProject;
grant create table to NProject;
grant create trigger to NProject;
grant create view to NProject;
grant unlimited tablespace to NProject;
```

Приложение Е

Листинг генерации структуры БД

```
CREATE TABLE CONTRACT
(
  ContractId      NUMBER NOT NULL ,
  ConNum          VARCHAR2(100) NULL ,
  ConDate         DATE NULL ,
  BeginDate       DATE NULL ,
  EndDate         DATE NULL
);

CREATE UNIQUE INDEX XPCONTRACT ON CONTRACT (ContractId ASC);
ALTER TABLE CONTRACT ADD CONSTRAINT XPCONTRACT PRIMARY KEY (ContractId);

CREATE TABLE CONTRACTSERVICE
(
  ContractId      NUMBER NOT NULL ,
  ServiceId       NUMBER NOT NULL ,
  ContractServiceId NUMBER NOT NULL
);

CREATE UNIQUE INDEX XPCONTRACTSERVICE ON CONTRACTSERVICE (ContractServiceId ASC);
ALTER TABLE CONTRACTSERVICE ADD CONSTRAINT XPCONTRACTSERVICE PRIMARY KEY (ContractServiceId);

CREATE TABLE DEP
(
  DepId           NUMBER NOT NULL ,
  DepName         VARCHAR2(100) NULL ,
  OrgId           NUMBER NOT NULL
);

CREATE UNIQUE INDEX XPDEP ON DEP(DepId ASC);
ALTER TABLE DEP ADD CONSTRAINT XPDEP PRIMARY KEY (DepId);

CREATE TABLE DIVISION
(
  DivisionId      NUMBER NOT NULL ,
  DivisionName    VARCHAR2(100) NULL ,
  DepId           NUMBER NOT NULL
);

CREATE UNIQUE INDEX XPDIVISION ON DIVISION(DivisionId ASC);
ALTER TABLE DIVISION ADD CONSTRAINT XPDIVISION PRIMARY KEY (DivisionId);

CREATE TABLE EMP
(
  EmpId           NUMBER NOT NULL ,
  Family          VARCHAR2(100) NULL ,
  Name            VARCHAR2(100) NULL ,
  Patonymic      VARCHAR2(100) NULL ,
  Phone           VARCHAR2(30) NULL ,
  EMail           VARCHAR2(100) NULL ,
  OracleName      VARCHAR2(100) NULL ,
  IsAdmin         VARCHAR2(1) NULL
);

CREATE UNIQUE INDEX XPEMP ON EMP (EmpId ASC);
ALTER TABLE EMP ADD CONSTRAINT XPEMP PRIMARY KEY (EmpId);

CREATE TABLE INITIATOR
```

```
(
  InitiatorId      NUMBER NOT NULL ,
  InitiatorNum     VARCHAR2(50) NULL ,
  Family           VARCHAR2(100) NULL ,
  Name             VARCHAR2(100) NULL ,
  Patronymic       VARCHAR2(100) NULL ,
  Status           VARCHAR2(100) NULL ,
  Phone            VARCHAR2(30) NULL ,
  Email            VARCHAR2(100) NULL ,
  DivisionId       NUMBER NOT NULL ,
  PostId           NUMBER NOT NULL
);
```

```
CREATE UNIQUE INDEX XPINITIATOR ON INITIATOR (InitiatorId ASC);
ALTER TABLE INITIATOR ADD CONSTRAINT XPINITIATOR PRIMARY KEY (InitiatorId);
```

```
CREATE TABLE ORG
```

```
(
  OrgId           NUMBER NOT NULL ,
  OrgName         VARCHAR2(100) NULL ,
  OgrAdr1         VARCHAR2(1000) NULL ,
  OrgAdr2         VARCHAR2(1000) NULL ,
  INN             NUMBER NULL
);
```

```
CREATE UNIQUE INDEX XPKORG ON ORG (OrgId ASC);
ALTER TABLE ORG ADD CONSTRAINT XPKORG PRIMARY KEY (OrgId);
```

```
CREATE TABLE POST
```

```
(
  PostId          NUMBER NOT NULL ,
  PostName        VARCHAR2(1000) NULL
);
```

```
CREATE UNIQUE INDEX XPKPOST ON POST (PostId ASC);
ALTER TABLE POST ADD CONSTRAINT XPKPOST PRIMARY KEY (PostId);
```

```
CREATE TABLE REPORT
```

```
(
  ReportId        NUMBER NOT NULL ,
  InitiatorId     NUMBER NOT NULL ,
  EmpId           NUMBER NOT NULL ,
  Theme           VARCHAR2(100) NULL ,
  Description      VARCHAR2(1000) NULL ,
  Priority         NUMBER NULL ,
  RepTime         TIMESTAMP NULL ,
  RepDate         DATE NULL ,
  ContractServiceId NUMBER NOT NULL ,
  Status          NUMBER NULL
);
```

```
CREATE UNIQUE INDEX XPKREPORT ON REPORT (ReportId ASC);
ALTER TABLE REPORT ADD CONSTRAINT XPKREPORT PRIMARY KEY (ReportId);
```

```
CREATE TABLE SERVICE
```

```
(
  ServiceId       NUMBER NOT NULL ,
  Name            VARCHAR2(100) NULL
);
```

```
CREATE UNIQUE INDEX XPKSERVICE ON SERVICE (ServiceId ASC);
ALTER TABLE SERVICE ADD CONSTRAINT XPKSERVICE PRIMARY KEY (ServiceId);
```

```

--FOREIGN KEYS
ALTER TABLE CONTRACTSERVICE ADD (CONSTRAINT R_6 FOREIGN KEY (ContractId) REFERENCES
CONTRACT (ContractId));
ALTER TABLE CONTRACTSERVICE ADD (CONSTRAINT R_7 FOREIGN KEY (ServiceId) REFERENCES
SERVICE (ServiceId));
ALTER TABLE DEP ADD (CONSTRAINT R_1 FOREIGN KEY (OrgId) REFERENCES ORG (OrgId));
ALTER TABLE DIVISION ADD (CONSTRAINT R_2 FOREIGN KEY (DepId) REFERENCES DEP (DepId));
ALTER TABLE INITIATOR ADD (CONSTRAINT R_3 FOREIGN KEY (DivisionId) REFERENCES DIVISION
(DivisionId));
ALTER TABLE INITIATOR ADD (CONSTRAINT R_9 FOREIGN KEY (PostId) REFERENCES POST (PostId));
ALTER TABLE REPORT ADD (CONSTRAINT R_4 FOREIGN KEY (InitiatorId) REFERENCES INITIATOR
(InitiatorId));
ALTER TABLE REPORT ADD (CONSTRAINT R_5 FOREIGN KEY (EmpId) REFERENCES EMP (EmpId));
ALTER TABLE REPORT ADD (CONSTRAINT R_8 FOREIGN KEY (ContractServiceId) REFERENCES
CONTRACTSERVICE (ContractServiceId));

-- Create sequence
create sequence GenId
start with 1
increment by 1;

```

Приложение Ж

Листинг программы

```
unit MainForm;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, cxGraphics, cxControls, cxLookAndFeels, cxLookAndFeelPainters,
  cxStyles, cxCustomData, cxFilter, cxData, cxDataStorage, cxEdit, DB, cxDBData,
  cxGridCustomTableView, cxGridTableView, cxGridDBTableView, cxGridLevel,
  cxClasses, cxGridCustomView, cxGrid, IBCustomDataSet, IBTable, IBDatabase,
  ActnList, dxBar, ExtCtrls, ImgList, OracleData, Oracle, dxStatusBar;

type
  TfrmMain = class(TForm)
    dxBarManager: TdxBarManager;
    bMain: TdxBar;
    dxBarDockControl1: TdxBarDockControl;
    dxBarButton1: TdxBarButton;
    ActionList: TActionList;
    acCodificator: TAction;
    pMain: TPanel;
    Images: TImageList;
    TaskImages: TImageList;
    acReport: TAction;
    acExit: TAction;
    dxBarButton3: TdxBarButton;
    acContract: TAction;
    dxBarButton4: TdxBarButton;
    dxBarButton5: TdxBarButton;
    acReports: TAction;
    dxBarButton2: TdxBarButton;
    StatusBar: TdxStatusBar;
    procedure FormCreate(Sender: TObject);
    procedure acCodificatorExecute(Sender: TObject);
    procedure acReportExecute(Sender: TObject);
    procedure acExitExecute(Sender: TObject);
    procedure acContractExecute(Sender: TObject);
    procedure acReportsExecute(Sender: TObject);
  private
    FTaskForm: TForm; //форма текущей задачи
    procedure TuneTaskForm; //настройка формы текущей задачи
    procedure TuneTaskBar(AAction: TAction); //настройка панели задач
  end;

var
  frmMain: TfrmMain;

implementation

uses CodificatorForm, nvtUtils, ReportForm, ContractForm, DataModule,
  ReportsForm, StrUtils;

{$R *.dfm}

//запуск задачи ДОГОВОРА
procedure TfrmMain.acContractExecute(Sender: TObject);
begin
  FTaskForm.Free; // освобождаем форму текущей задачи
  FTaskForm := TfrmContract.Create(nil); //создаем форму задачи как форму договоров
```

```

TuneTaskForm; //настраиваем форму
TuneTaskBar(Sender as TAction); //настраиваем командную панель
end;

procedure TfrmMain.acReportsExecute(Sender: TObject);
begin
  FTaskForm.Free;
  FTaskForm := TfrmReport.Create(nil);
  TuneTaskForm;
  TuneTaskBar(Sender as TAction);
end;

procedure TfrmMain.acCodificatorExecute(Sender: TObject);
begin
  FTaskForm.Free;
  FTaskForm := TfrmCodificator.Create(nil);
  TuneTaskForm;
  TuneTaskBar(Sender as TAction);
end;

//выход из программы
procedure TfrmMain.acExitExecute(Sender: TObject);
begin
  if ConfirmDlg('Вы действительно хотите закрыть приложение?') then
    Application.Terminate;
end;

procedure TfrmMain.acReportExecute(Sender: TObject);
begin
  FTaskForm.Free;
  FTaskForm := TfrmReports.Create(nil);
  TuneTaskForm;
  TuneTaskBar(Sender as TAction);
end;

//отрабатывает при создании формы
procedure TfrmMain.FormCreate(Sender: TObject);
begin
  InitFormState(Self); //настройка формы
  StatusBar.Panels[0].Text := DM.FOraName; //Oracle-имя пользователя - в статус бар
  StatusBar.Panels[1].Text := 'Режим ' + IfThen(DM.FIsAdmin, 'администратора', 'пользователя'); //тип
  пользователя - в статус бар
  acReport.Execute; //по умолчанию отображаем форму обращений
  acContract.Visible := DM.FIsAdmin; //настройка видимости задачи ДОГОВОРА
  acCodificator.Visible := DM.FIsAdmin; //настройка видимости задачи КОДИФИКАТОРЫ
end;

//процедура настройки панели задач
procedure TfrmMain.TuneTaskBar(AAction: TAction);
var
  i: Integer;
  vLink: TdxBarItemLink; //ссылка на кнопку панели
begin
  for i := 0 to bMain.ItemLinks.Count - 1 do //идем по всем ссылкам
    begin
      vLink := bMain.ItemLinks[i]; //получаем ссылку (исключительно для удобочитаемости кода сделано)
      if vLink.Control is TdxBarButtonControl then //проверяем ссылку на принадлежность к кнопкам
        TdxBarButton(vLink.Item).Lowered := TdxBarButton(vLink.Item).Action = AAction; //если надо - делаем
        кнопку нажатой
      end;
    end;
  end;

  //процедура настройки формы активной задачи

```

```

procedure TfrmMain.TuneTaskForm;
begin
  FTaskForm.BorderStyle := bsNone; //скрываем границы формы (шапку в том числе)
  FTaskForm.Align := alClient; // форма занимает все поверхность панели pMain
  FTaskForm.FormStyle := fsNormal;
  FTaskForm.Parent := pMain; //форма лежит внутри главной формы на панели pMain
  FTaskForm.Show; //показываем форму пользователю
end;

end.

unit CodificatorForm;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, cxGraphics, cxControls, cxLookAndFeels,
  cxLookAndFeelPainters, cxCustomData, cxStyles, cxTL, cxTextEdit,
  cxTLdxBarBuiltInMenu, cxInplaceContainer, ActnList, dxBar, ExtCtrls, cxClasses,
  DB, cxFilter, cxData, cxDataStorage, cxEdit, cxDBData, cxGridLevel,
  cxGridCustomView, cxGridCustomTableView, cxGridTableView, cxGridDBTableView,
  cxGrid, IBCustomDataSet, IBTable, nvtCaptionPanel, cxSplitter, cxContainer,
  cxMaskEdit, cxDropDownEdit, cxLookupEdit, cxDBLookupEdit, cxDBLookupComboBox,
  OracleData, cxPC;

type
  TfrmCodificator = class(TForm)
    tlCodificator: TcxTreeList;
    cxTreeList1Column1: TcxTreeListColumn;
    cxStyleRepository1: TcxStyleRepository;
    cxStyle1: TcxStyle;
    dxBarManager1: TdxBarManager;
    dxBarManager1Bar1: TdxBar;
    pMain: TPanel;
    dxBarDockControl1: TdxBarDockControl;
    ActionList: TActionList;
    acAdd: TAction;
    acEdit: TAction;
    acDel: TAction;
    dxBarButton1: TdxBarButton;
    dxBarButton2: TdxBarButton;
    dxBarButton3: TdxBarButton;
    DS: TDataSource;
    tvCodificator: TcxGridDBTableView;
    cxGrid1Level1: TcxGridLevel;
    cxGrid1: TcxGrid;
    pCaption: TNvtCaptionPanel;
    cxSplitter1: TcxSplitter;
    pCodif: TNvtCaptionPanel;
    pCod: TPanel;
    dInitiator: TOracleDataSet;
    dInitiatorINITIATORID: TFloatField;
    dInitiatorINITIATORNUM: TStringField;
    dInitiatorFAMILY: TStringField;
    dInitiatorNAME: TStringField;
    dInitiatorPATRONYMIC: TStringField;
    dInitiatorSTATUS: TStringField;
    dInitiatorPHONE: TStringField;
    dInitiatorEMAIL: TStringField;
    dInitiatorDIVISIONID: TFloatField;
    dInitiatorPOSTID: TFloatField;
    dDep: TOracleDataSet;
  end;

```

```

dDivision: TOracleDataSet;
dOrg: TOracleDataSet;
dPost: TOracleDataSet;
dService: TOracleDataSet;
dOrgORGID: TFloatField;
dOrgORGNAME: TStringField;
dOrgOGRADR1: TStringField;
dOrgORGADR2: TStringField;
dOrgINN: TFloatField;
dPostPOSTID: TFloatField;
dPostPOSTNAME: TStringField;
dServiceSERVICEID: TFloatField;
dServiceNAME: TStringField;
dDepDEPID: TFloatField;
dDepDEPNAME: TStringField;
dDepORGID: TFloatField;
dDepORG: TStringField;
dDivisionDIVISIONID: TFloatField;
dDivisionDIVISIONNAME: TStringField;
dDivisionDEPID: TFloatField;
dDivisionDEP: TStringField;
dInitStatus: TOracleDataSet;
dInitiatorStatusLkp: TStringField;
dInitStatusSTATUS: TStringField;
dInitiatorDivision: TStringField;
dInitiatorPost: TStringField;
pOrg: TPanel;
sOrg: TcxSplitter;
cxGrid2: TcxGrid;
cxGridDBTableView1: TcxGridDBTableView;
cxGridLevel1: TcxGridLevel;
dsDep: TDataSource;
dsDivision: TDataSource;
dxBarManager1Bar2: TdxBar;
dxBarManager1Bar3: TdxBar;
dxBarDockControl2: TdxBarDockControl;
cxSplitter2: TcxSplitter;
pDivision: TPanel;
dxBarDockControl3: TdxBarDockControl;
cxGrid3: TcxGrid;
cxGridDBTableView2: TcxGridDBTableView;
cxGridLevel2: TcxGridLevel;
cxGridDBTableView1DEPID: TcxGridDBCColumn;
cxGridDBTableView1DEPNAME: TcxGridDBCColumn;
cxGridDBTableView1ORGID: TcxGridDBCColumn;
cxGridDBTableView1ORG: TcxGridDBCColumn;
cxGridDBTableView2DIVISIONID: TcxGridDBCColumn;
cxGridDBTableView2DIVISIONNAME: TcxGridDBCColumn;
cxGridDBTableView2DEPID: TcxGridDBCColumn;
cxGridDBTableView2DEP: TcxGridDBCColumn;
acNewDep: TAction;
acEditDep: TAction;
acDelDep: TAction;
acNewDiv: TAction;
acEditDiv: TAction;
acDelDiv: TAction;
dxBarButton4: TdxBarButton;
dxBarButton5: TdxBarButton;
dxBarButton6: TdxBarButton;
dxBarButton7: TdxBarButton;
dxBarLargeButton1: TdxBarLargeButton;
dxBarButton8: TdxBarButton;
dxBarSubItem1: TdxBarSubItem;

```

```

dxBarButton9: TdxBarButton;
dOrg_Lkp: TOracleDataSet;
FloatField1: TFloatField;
StringField1: TStringField;
dDep_Lkp: TOracleDataSet;
FloatField3: TFloatField;
StringField4: TStringField;
dPost_Lkp: TOracleDataSet;
FloatField5: TFloatField;
StringField6: TStringField;
dDiv_Lkp: TOracleDataSet;
FloatField6: TFloatField;
StringField7: TStringField;
procedure ActionListUpdate(Action: TBasicAction; var Handled: Boolean);
procedure FormCreate(Sender: TObject);
procedure tlCodificatorFocusedNodeChanged(Sender: TcxCustomTreeList;
  APrevFocusedNode, AFocusedNode: TcxTreeNode);
procedure acEditExecute(Sender: TObject);
procedure acDelExecute(Sender: TObject);
procedure acAddExecute(Sender: TObject);
procedure tvCodificatorCellClick(Sender: TcxCustomGridTableView;
  ACellViewInfo: TcxGridTableDataCellViewInfo; AButton: TMouseButton;
  AShift: TShiftState; var AHandled: Boolean);
procedure acNewDepExecute(Sender: TObject);
procedure acEditDepExecute(Sender: TObject);
procedure acDelDepExecute(Sender: TObject);
procedure acNewDivExecute(Sender: TObject);
procedure acEditDivExecute(Sender: TObject);
procedure acDelDivExecute(Sender: TObject);
procedure cxGridDBTableView1CellClick(Sender: TcxCustomGridTableView;
  ACellViewInfo: TcxGridTableDataCellViewInfo; AButton: TMouseButton;
  AShift: TShiftState; var AHandled: Boolean);
procedure cxGridDBTableView2CellClick(Sender: TcxCustomGridTableView;
  ACellViewInfo: TcxGridTableDataCellViewInfo; AButton: TMouseButton;
  AShift: TShiftState; var AHandled: Boolean);
private
  FTable: TOracleDataSet; //датасет активного кодификатора
  procedure ShowData;
  function GetCodificatorCaption(AReadOnly: Boolean = False): string; //название активного справочника

  procedure AddRecord(ATable: TOracleDataSet; ACodName: string); //добавляет запись в ATable
  procedure DelRecord(ATable: TOracleDataSet; ACodName: string); //удаляет запись из ATable
  procedure EditRecord(ATable: TOracleDataSet; ACodName: string); //редактирует запись ATable
end;

var
  frmCodificator: TfrmCodificator;

implementation

uses nvtUtils, DSEditForm, MainForm, DataModule;

{$R *.dfm}

procedure TfrmCodificator.acAddExecute(Sender: TObject);
begin
  AddRecord(FTable, GetCodificatorCaption);
end;

procedure TfrmCodificator.acDelDepExecute(Sender: TObject);
begin
  DelRecord(dDep, 'Департамент');
end;

```

```

procedure TfrmCodificator.acDelDivExecute(Sender: TObject);
begin
    DelRecord(dDivision, 'Отдел');
end;

procedure TfrmCodificator.acDelExecute(Sender: TObject);
begin
    DelRecord(FTable, GetCodificatorCaption);
end;

procedure TfrmCodificator.acEditDepExecute(Sender: TObject);
begin
    EditRecord(dDep, 'Департамент');
end;

procedure TfrmCodificator.acEditDivExecute(Sender: TObject);
begin
    EditRecord(dDivision, 'Отдел');
end;

procedure TfrmCodificator.acEditExecute(Sender: TObject);
begin
    EditRecord(FTable, GetCodificatorCaption);
end;

procedure TfrmCodificator.acNewDepExecute(Sender: TObject);
begin
    AddRecord(dDep, 'Департамент');
end;

procedure TfrmCodificator.acNewDivExecute(Sender: TObject);
begin
    AddRecord(dDivision, 'Отдел');
end;

procedure TfrmCodificator.ActionListUpdate(Action: TBasicAction;
    var Handled: Boolean);
begin
    acEdit.Enabled := not FTable.IsEmpty;
    acDel.Enabled := acEdit.Enabled;
    //департаменты
    acNewDep.Enabled := not dOrg.IsEmpty;
    acEditDep.Enabled := not dDep.IsEmpty;
    acDelDep.Enabled := acEditDep.Enabled;
    //отделы
    acNewDiv.Enabled := not dDep.IsEmpty;
    acEditDiv.Enabled := not dDivision.IsEmpty;
    acDelDiv.Enabled := acEditDiv.Enabled;
end;

procedure TfrmCodificator.AddRecord(ATable: TOracleDataSet; ACodName: string);
begin
    ATable.Insert;
    if ShowDatasetEditDlg(ATable, Format('Создание: %s', [ACodName])) then
        dm.OracleSession.ApplyUpdates([ATable], True)
    else
        dm.OracleSession.CancelUpdates([ATable])
end;

procedure TfrmCodificator.cxGridDBTableView1CellClick(
    Sender: TcxCustomGridTableView; ACellViewInfo: TcxGridTableDataCellViewInfo;
    AButton: TMouseButton; AShift: TShiftState; var AHandled: Boolean);

```

```

begin
  if Sender.Controller.IsDbClick then
    begin
      acEditDep.Execute;
      AHandled := True;
    end;
  end;
end;

procedure TfrmCodificator.cxGridDBTableView2CellClick(
  Sender: TcxCustomGridTableView; ACellViewInfo: TcxGridTableDataCellViewInfo;
  AButton: TMouseButton; AShift: TShiftState; var AHandled: Boolean);
begin
  if Sender.Controller.IsDbClick then
    begin
      acEditDiv.Execute;
      AHandled := True;
    end;
  end;
end;

procedure TfrmCodificator.DelRecord(ATable: TOracleDataSet; ACodName: string);
begin
  if not ConfirmDlg(Format('Действительно удалить запись кодификатора "%s"?', [ACodName])) then
    Exit;

  ATable.Delete;
  dm.OracleSession.ApplyUpdates([ATable], True)
end;

procedure TfrmCodificator.EditRecord(ATable: TOracleDataSet; ACodName: string);
begin
  ATable.Edit;
  if ShowDatasetEditDlg(ATable, Format('Редактирование: %s', [ACodName])) then
    dm.OracleSession.ApplyUpdates([ATable], True)
  else
    dm.OracleSession.CancelUpdates([ATable])
end;

procedure TfrmCodificator.FormCreate(Sender: TObject);
begin
  InitFormState(Self);
  FTable := dPost;
  tlCodificator.Root.getFirstChild.Focused := True;
end;

function TfrmCodificator.GetCodificatorCaption(AReadOnly: Boolean): string;
begin
  if Assigned(tlCodificator.FocusedNode) then
    Result := tlCodificator.FocusedNode.Texts[0];
  if AReadOnly then
    Result := Format('%s (только чтение)', [Result]);
end;

//настраивает текущий справочник
procedure TfrmCodificator.ShowData;
begin
  pCaption.Caption := Format('Текущий кодификатор: "%s"', [GetCodificatorCaption]); //надпись
  DS.DataSet := FTable; //настройка датасорса
  tvCodificator.ClearItems; //очищаем колонки грида
  tvCodificator.DataController.CreateAllItems; //создаем колонки грида заново
  FTable.Open; //открываем таблицу кодификатора
  if FTable = dOrg then //если текущий кодификатор - организации, то открываем связанные таблицы
    департаментов и отделов
  begin

```

```

    dDep.Open;
    dDivision.Open;
end;
end;

//обработка пользовательской навигации по панели кодификаторов
procedure TfrmCodificator.tlCodificatorFocusedNodeChanged(
    Sender: TcxCustomTreeList; APrevFocusedNode, AFocusedNode: TcxTreeNode);
begin
    if not Assigned(FTable) then //исключаем возможные "левые" отработки процедуры
        Exit;

    FTable.Close; //закрываем датасет текущего кодификатора
    case tlCodificator.FocusedNode.Index of //назначаем датасет текущего кодификатора
        0: FTable := dPost;
        1: FTable := dInitiator;
        2: FTable := dOrg;
        3: FTable := dService;
    end;
    ShowData; //показываем текущий кодификатор

    //для справочника организаций дополнительно показываем панель департаментов и отделов
    pOrg.Visible := tlCodificator.FocusedNode.Index = 2;
    sOrg.Visible := pOrg.Visible;
end;

procedure TfrmCodificator.tvCodificatorCellClick(Sender: TcxCustomGridView;
    ACellViewInfo: TcxGridViewDataCellViewInfo; AButton: TMouseButton;
    AShift: TShiftState; var AHandled: Boolean);
begin
    if Sender.Controller.IsDoubleClick then
        begin
            acEdit.Execute;
            AHandled := True;
        end;
end;

end.

unit ContractForm;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, DB, IBCustomDataSet, IBTable, cxGraphics, cxControls, cxLookAndFeels,
    cxLookAndFeelPainters, cxStyles, cxCustomData, cxFilter, cxData,
    cxDataStorage, cxEdit, cxDBData, cxGridCustomView, cxGridCustomTableView,
    cxGridTableView, cxGridBandedTableView, cxGridDBBandedTableView, cxClasses,
    cxGridLevel, cxGrid, dxBar, ActnList, nvtCaptionPanel, IBQuery, dxBarExtItems,
    cxGridChartView, cxGridDBChartView, ExtCtrls, cxSplitter, cxImageComboBox,
    OracleData;

type
    TfrmContract = class(TForm)
        ds: TDataSource;
        dxBarManager: TdxBarManager;
        ActionList: TActionList;
        pCaption: TNvtCaptionPanel;
        cxStyleRepository: TcxStyleRepository;
        cxStyle1: TcxStyle;
        dxBarManager1Bar1: TdxBar;
    end;

```

```

dxBarDockControl1: TdxBarDockControl;
eYear: TdxBarSpinEdit;
acPrev: TAction;
acNext: TAction;
dContract: TOracleDataSet;
cxGrid2: TcxGrid;
cxGridDBBandedTableView1: TcxGridDBBandedTableView;
cxGridLevel1: TcxGridLevel;
acNew: TAction;
acEdit: TAction;
acDel: TAction;
dxBarButton1: TdxBarButton;
dxBarButton4: TdxBarButton;
dxBarButton5: TdxBarButton;
dContractCONTRACTID: TFloatField;
dContractCONNUM: TStringField;
dContractCONDATE: TDateTimeField;
dContractBEGINDATE: TDateTimeField;
dContractENDDATE: TDateTimeField;
cxGridDBBandedTableView1CONTRACTID: TcxGridDBBandedColumn;
cxGridDBBandedTableView1CONNUM: TcxGridDBBandedColumn;
cxGridDBBandedTableView1CONDATE: TcxGridDBBandedColumn;
cxGridDBBandedTableView1BEGINDATE: TcxGridDBBandedColumn;
cxGridDBBandedTableView1ENDDATE: TcxGridDBBandedColumn;
dContractService: TOracleDataSet;
dContractServiceCONTRACTID: TFloatField;
dContractServiceSERVICEID: TFloatField;
dContractServiceCONTRACTSERVICEID: TFloatField;
acNewSer: TAction;
acEditSer: TAction;
acDelSer: TAction;
Panel1: TPanel;
cxGrid1: TcxGrid;
cxGridDBBandedTableView2: TcxGridDBBandedTableView;
cxGridLevel2: TcxGridLevel;
NvtCaptionPanel1: TNvtCaptionPanel;
dxBarDockControl2: TdxBarDockControl;
dxBarManagerBar1: TdxBar;
dxBarButton6: TdxBarButton;
dxBarButton7: TdxBarButton;
dxBarButton8: TdxBarButton;
dsDetail: TDataSource;
dService: TOracleDataSet;
dServiceSERVICEID: TFloatField;
dServiceNAME: TStringField;
dContractServiceService: TStringField;
cxGridDBBandedTableView2CONTRACTID: TcxGridDBBandedColumn;
cxGridDBBandedTableView2SERVICEID: TcxGridDBBandedColumn;
cxGridDBBandedTableView2CONTRACTSERVICEID: TcxGridDBBandedColumn;
cxGridDBBandedTableView2Service: TcxGridDBBandedColumn;
procedure FormCreate(Sender: TObject);
procedure eMonthChange(Sender: TObject);
procedure eYearCurChange(Sender: TObject);
procedure acNewExecute(Sender: TObject);
procedure acEditExecute(Sender: TObject);
procedure acDelExecute(Sender: TObject);
procedure ActionListUpdate(Action: TBasicAction; var Handled: Boolean);
procedure acNewSerExecute(Sender: TObject);
procedure acEditSerExecute(Sender: TObject);
procedure acDelSerExecute(Sender: TObject);
procedure FormDestroy(Sender: TObject);
private
FUpdate: Boolean;

```

```

function GetCurDate: TDate;
procedure RefreshData;
end;

var
  frmContract: TfrmContract;

implementation

uses DSEditForm, nvtUtils, DateUtils, Math, DataModule;

{$R *.dfm}

procedure TfrmContract.acDelExecute(Sender: TObject);
begin
  if not ConfirmDlg('Действительно удалить запись?') then
    Exit;
  dContract.Delete;
  dm.OracleSession.ApplyUpdates([dContract], True)
end;

procedure TfrmContract.acDelSerExecute(Sender: TObject);
begin
  if not ConfirmDlg('Действительно удалить запись?') then
    Exit;
  dContractService.Delete;
  dm.OracleSession.ApplyUpdates([dContractService], True)
end;

procedure TfrmContract.acEditExecute(Sender: TObject);
begin
  dContract.Edit;
  if ShowDatasetEditDlg(dContract, 'Редактирование договора') then
    dm.OracleSession.ApplyUpdates([dContract], True)
  else
    dm.OracleSession.CancelUpdates([dContract])
end;

procedure TfrmContract.acEditSerExecute(Sender: TObject);
begin
  dContractService.Edit;
  if ShowDatasetEditDlg(dContractService, 'Редактирование услуги договора') then
    dm.OracleSession.ApplyUpdates([dContractService], True)
  else
    dm.OracleSession.CancelUpdates([dContractService])
end;

procedure TfrmContract.acNewExecute(Sender: TObject);
begin
  dContract.Insert;
  if ShowDatasetEditDlg(dContract, 'Создание договора') then
    dm.OracleSession.ApplyUpdates([dContract], True)
  else
    dm.OracleSession.CancelUpdates([dContract])
end;

procedure TfrmContract.acNewSerExecute(Sender: TObject);
begin
  dContractService.Insert;
  if ShowDatasetEditDlg(dContractService, 'Создание услуги договора') then
    dm.OracleSession.ApplyUpdates([dContractService], True)
  else
    dm.OracleSession.CancelUpdates([dContractService])
end;

```

```

end;

procedure TfrmContract.ActionListUpdate(Action: TBasicAction;
  var Handled: Boolean);
begin
  acNew.Enabled := dm.FIsAdmin;
  acEdit.Enabled := dm.FIsAdmin and not dContract.IsEmpty;
  acDel.Enabled := dm.FIsAdmin and not dContract.IsEmpty;

  acNewSer.Enabled := dm.FIsAdmin and not dContract.IsEmpty;
  acEditSer.Enabled := dm.FIsAdmin and not dContractService.IsEmpty;
  acDelSer.Enabled := dm.FIsAdmin and not dContractService.IsEmpty;
end;

procedure TfrmContract.eMonthChange(Sender: TObject);
begin
  if FUpdate then
    Exit;
  RefreshData;
end;

procedure TfrmContract.eYearCurChange(Sender: TObject);
begin
  eYear.IntValue := eYear.IntCurValue;
end;

procedure TfrmContract.FormCreate(Sender: TObject);
begin
  InitFormState(Self);
  dContractService.Open;
  FUpdate := True;
  eYear.Value := RestoreParam(Self, 'Year', YearOf(Now));
  FUpdate := False;
  RefreshData;
end;

procedure TfrmContract.FormDestroy(Sender: TObject);
begin
  StoreParam(Self, 'Year', eYear.IntValue);
end;

function TfrmContract.GetCurDate: TDate;
begin
  Result := EncodeDate(eYear.IntValue, 1, 1);
end;

procedure TfrmContract.RefreshData;
begin
  dContract.Close;
  dContract.SetVariable(0, GetCurDate);
  dContract.SetVariable(1, IncMonth(GetCurDate, 12));
  dContract.Open;
end;

end.

unit DataModule;

interface

uses
  SysUtils, Classes, Oracle;

```

```

type
  TDM = class(TDataModule)
    OracleSession: TOracleSession;
    OracleLogon: TOracleLogon;
    procedure DataModuleCreate(Sender: TObject);
  public
    FUserId: Integer;
    FOraName: string;
    FIsAdmin: Boolean;
  end;

var
  DM: TDM;

implementation

{$R *.dfm}

uses dialogs, Forms;

procedure TDM.DataModuleCreate(Sender: TObject);
begin
  try
    OracleLogon.Execute; //коннект к БД
  except //блок ниже отрабатываем если по каким то причинам коннект к БД не состоялся
    if not OracleSession.Connected then //если не получилось, то говорим об этом пользователю
      ShowMessage('Не установлено соединение с БД!')
    else
      ShowMessage('Ошибка запуска!');
    Application.Terminate; //закрываем программу
  end;
end;

end.

unit LoginForm;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, cxGraphics, cxControls, cxLookAndFeels,
  cxLookAndFeelPainters, cxContainer, cxEdit, cxTextEdit, Oracle, DB, OracleData;

type
  TfrmLogin = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Button1: TButton;
    Button2: TButton;
    Label1: TLabel;
    Label2: TLabel;
    eName: TcxTextEdit;
    ePas: TcxTextEdit;
    dEmp: TOracleDataSet;
    dEmpEMPID: TFloatField;
    dEmpORACLENAME: TStringField;
    dEmpORACLEPAS: TStringField;
    dEmpISADMIN: TStringField;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  end;

```

```

public
{ Public declarations }
end;

function UserLogin(out AUserId: Integer; out AOraName: string; out AIsAdmin: Boolean): Boolean;

var
    frmLogin: TfrmLogin;

implementation

{$R *.dfm}

uses nvtUtils;

function UserLogin(out AUserId: Integer; out AOraName: string; out AIsAdmin: Boolean): Boolean;
begin
    with TfrmLogin.Create(nil) do
        try
            Result := ShowModal = mrOk;
            if Result then
                begin
                    AUserId := dEmpEMPID.AsInteger;
                    AOraName := dEmpORACLENAME.AsString;
                    AIsAdmin := dEmpISADMIN.Value = 'Y';
                end;
            finally
                Free;
            end;
        end;
    end;
end;

procedure TfrmLogin.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    if ModalResult = mrOk then
        begin
            dEmp.Open;
            if not (dEmp.Locate('OracleName', eName.Text, []) and
                SameText(dEmpORACLEPAS.AsString, ePas.Text)) then
                begin
                    InformDlg('Неверный логин/пароль', mtError);
                    Action := caNone;
                end;
            end;
        end;
end;

procedure TfrmLogin.FormCreate(Sender: TObject);
begin
    InitFormState(Self)
end;

end.

unit ReportsForm;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, DB, IBCustomDataSet, IBTable, cxGraphics, cxControls, cxLookAndFeels,
    cxLookAndFeelPainters, cxStyles, cxCustomData, cxFilter, cxData,
    cxDataStorage, cxEdit, cxDBData, cxGridCustomView, cxGridCustomTableView,
    cxGridTableView, cxGridBandedTableView, cxGridDBBandedTableView, cxClasses,

```

cxGridLevel, cxGrid, dxBar, ActnList, nvtCaptionPanel, IBQuery, dxBarExtItems, cxGridChartView, cxGridDBChartView, ExtCtrls, cxSplitter, cxImageComboBox, OracleData, dxSkinsCore, dxSkinBlack, dxSkinBlue, dxSkinCaramel, dxSkinCoffee, dxSkinDarkRoom, dxSkinDarkSide, dxSkinFoggy, dxSkinGlassOceans, dxSkiniMaginary, dxSkinLilian, dxSkinLiquidSky, dxSkinLondonLiquidSky, dxSkinMcSkin, dxSkinMoneyTwins, dxSkinOffice2007Black, dxSkinOffice2007Blue, dxSkinOffice2007Green, dxSkinOffice2007Pink, dxSkinOffice2007Silver, dxSkinPumpkin, dxSkinSeven, dxSkinSharp, dxSkinSilver, dxSkinSpringTime, dxSkinStardust, dxSkinSummer2008, dxSkinsDefaultPainters, dxSkinValentine, dxSkinXmas2008Blue, dxSkinscxPCPainter, dxSkinsdxBarPainter;

type

```
TfrmReports = class(TForm)
  ds: TDataSource;
  dxBarManager1: TdxBarManager;
  ActionList: TActionList;
  pCaption: TNvtCaptionPanel;
  cxStyleRepository1: TcxStyleRepository;
  cxStyle1: TcxStyle;
  dxBarManager1Bar1: TdxBar;
  dxBarDockControl1: TdxBarDockControl;
  eMonth: TdxBarCombo;
  eYear: TdxBarSpinEdit;
  acPrev: TAction;
  acNext: TAction;
  dxBarButton2: TdxBarButton;
  dxBarButton3: TdxBarButton;
  dReport: TOracleDataSet;
  dReportREPORTID: TFloatField;
  dReportINITIATORID: TFloatField;
  dReportEMPID: TFloatField;
  dReportTHEME: TStringField;
  dReportDESCRIPTION: TStringField;
  dReportPRIORITY: TFloatField;
  dReportREPPDATE: TDateTimeField;
  dReportCONTRACTSERVICEID: TFloatField;
  dReportSTATUS: TFloatField;
  dInitiator: TOracleDataSet;
  dInitiatorINITIATORID: TFloatField;
  dInitiatorFAMILY: TStringField;
  dInitiatorNAME: TStringField;
  dInitiatorPATRONYMIC: TStringField;
  cxGrid2: TcxGrid;
  cxGridDBBandedTableView1: TcxGridDBBandedTableView;
  cxGridLevel1: TcxGridLevel;
  dCS: TOracleDataSet;
  dCSCONTRACTSERVICEID: TFloatField;
  dCSCONNUM: TStringField;
  dCSNAME: TStringField;
  dReportInitiator: TStringField;
  dInitiatorFIO: TStringField;
  dReportContract: TStringField;
  cxGridDBBandedTableView1REPORTID: TcxGridDBBandedColumn;
  cxGridDBBandedTableView1INITIATORID: TcxGridDBBandedColumn;
  cxGridDBBandedTableView1EMPID: TcxGridDBBandedColumn;
  cxGridDBBandedTableView1THEME: TcxGridDBBandedColumn;
  cxGridDBBandedTableView1DESCRIPTION: TcxGridDBBandedColumn;
  cxGridDBBandedTableView1PRIORITY: TcxGridDBBandedColumn;
  cxGridDBBandedTableView1REPPDATE: TcxGridDBBandedColumn;
  cxGridDBBandedTableView1CONTRACTSERVICEID: TcxGridDBBandedColumn;
  cxGridDBBandedTableView1STATUS: TcxGridDBBandedColumn;
  cxGridDBBandedTableView1Initiator: TcxGridDBBandedColumn;
  cxGridDBBandedTableView1Contract: TcxGridDBBandedColumn;
```

```

dxBarButton1: TdxBarButton;
dxBarButton4: TdxBarButton;
dxBarButton5: TdxBarButton;
acnew: TAction;
acedit: TAction;
acdel: TAction;
dStatus_Lkp: TOracleDataSet;
dReportStatusStr: TStringField;
dxBarDockControl2: TdxBarDockControl;
dxBarManager1Bar2: TdxBar;
dStatus_LkpSTATUS: TFloatField;
dStatus_LkpSTATUSSTR: TStringField;
dReportREPTIME: TTimeField;
dEmp: TOracleDataSet;
dEmpEMPID: TFloatField;
dEmpFIO: TStringField;
dEmpFAMILY: TStringField;
dEmpNAME: TStringField;
dEmpPATONYMIC: TStringField;
dReportEmp: TStringField;
cxGridDBBandedTableView1Emp: TcxGridDBBandedColumn;
dPriority_Lkp: TOracleDataSet;
dReportPrStr: TStringField;
dPriority_LkpPRIORITY: TFloatField;
dPriority_LkpPRSTR: TStringField;
procedure FormCreate(Sender: TObject);
procedure eMonthChange(Sender: TObject);
procedure acPrevExecute(Sender: TObject);
procedure eYearCurChange(Sender: TObject);
procedure dInitiatorCalcFields(DataSet: TDataSet);
procedure acnewExecute(Sender: TObject);
procedure aceditExecute(Sender: TObject);
procedure acdelExecute(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure dEmpCalcFields(DataSet: TDataSet);
procedure ActionListUpdate(Action: TBasicAction; var Handled: Boolean);
procedure cxGridDBBandedTableView1CellClick(Sender: TcxCustomGridTableView;
  ACellViewInfo: TcxGridTableDataCellViewInfo; AButton: TMouseButton;
  AShift: TShiftState; var AHandled: Boolean);
procedure cxGridDBBandedTableView1CustomDrawCell(
  Sender: TcxCustomGridTableView; ACanvas: TcxCanvas;
  AViewInfo: TcxGridTableDataCellViewInfo; var ADone: Boolean);
private
  FUpdate: Boolean; //флаг для проверки не идут ли изменения
  function GetCurDate: TDate; //получение текущей даты, установленной пользователем
  procedure RefreshData; //обновление задачи
end;

var
  frmReports: TfrmReports;

implementation

uses DSEditForm, nvtUtils, DateUtils, Math, DataModule;

{$R *.dfm}

//удаление записи
procedure TfrmReports.acdelExecute(Sender: TObject);
begin
  if not ConfirmDlg('Действительно удалить запись?') then //спрашиваем пользователя
    Exit;
  dReport.Delete; //удаляем запись датасета

```

```

dm.OracleSession.ApplyUpdates([dReport], True) //сохраняем изменения в БД
end;

//редактирование записи
procedure TfrmReports.aceditExecute(Sender: TObject);
begin
  dReport.Edit; //ставим запись в режим редактирования
  if ShowDatasetEditDlg(dReport, 'Редактирование обращения') then
    dm.OracleSession.ApplyUpdates([dReport], True)
  else
    dm.OracleSession.CancelUpdates([dReport]); //отмена изменений в БД
end;

//создание записи
procedure TfrmReports.acnewExecute(Sender: TObject);
begin
  dReport.Insert; //добавляем запись в датасет
  if not DM.FIsAdmin then //если пользователь не админ, то автоматически прописываем исполнителя
    dReportEMPID.AsInteger := DM.FUserId;
  if ShowDatasetEditDlg(dReport, 'Создание обращения') then
    dm.OracleSession.ApplyUpdates([dReport], True)
  else
    dm.OracleSession.CancelUpdates([dReport])
end;

//навигация по периоду
procedure TfrmReports.acPrevExecute(Sender: TObject);
var
  vDelta: Integer; //для инкремента месяца
begin
  vDelta := IfThen(Sender = acPrev, -1, 1);
  FUpdate := True;
  try
    eMonth.ItemIndex := MonthOf(IncMonth(GetCurDate, vDelta)) - 1; //меняем значения контролов дат
    eYear.IntValue := YearOf(IncMonth(GetCurDate, vDelta));
  finally
    FUpdate := False;
    RefreshData; //обновляем задачу
  end;
end;

procedure TfrmReports.ActionListUpdate(Action: TBasicAction;
  var Handled: Boolean);
begin
  acedit.Enabled := not dReport.IsEmpty and (DM.FIsAdmin or (dReportEMPID.AsInteger = dm.FUserId));
  acdel.Enabled := acedit.Enabled;
end;

//по двойному клику по записи обращения переходит в режим редактирования
procedure TfrmReports.cxGridDBBandedTableView1CellClick(
  Sender: TcxCustomGridTableView; ACellViewInfo: TcxGridTableDataCellViewInfo;
  AButton: TMouseButton; AShift: TShiftState; var AHandled: Boolean);
begin
  if Sender.Controller.IsDbClick then
    begin
      acEdit.Execute;
      AHandled := True;
    end;
end;

procedure TfrmReports.cxGridDBBandedTableView1CustomDrawCell(
  Sender: TcxCustomGridTableView; ACanvas: TcxCanvas;
  AViewInfo: TcxGridTableDataCellViewInfo; var ADone: Boolean);

```

```

begin
  if AViewInfo.Selected then
    Exit;

  if AViewInfo.GridRecord.RecordIndex mod 2 = 0 then
    ACanvas.Brush.Color := clInfoBk
end;

//формирование Фамилия И.О.
procedure TfrmReports.dEmpCalcFields(DataSet: TDataSet);
begin
  dEmpFIO.Value := Format('%s %s.%s.', [dEmpFAMILY.AsString,
    Copy(dEmpFAMILY.AsString, 1, 1),
    Copy(dEmpPATONYMIC.AsString, 1, 1)]);
end;

procedure TfrmReports.dInitiatorCalcFields(DataSet: TDataSet);
begin
  dInitiatorFIO.Value := Format('%s %s.%s.', [dInitiatorFAMILY.AsString,
    Copy(dInitiatorNAME.AsString, 1, 1),
    Copy(dInitiatorPATRONYMIC.AsString, 1, 1)]);
end;

procedure TfrmReports.eMonthChange(Sender: TObject);
begin
  if FUpdate then
    Exit;
  RefreshData;
end;

procedure TfrmReports.eYearCurChange(Sender: TObject);
begin
  eYear.IntValue := eYear.IntCurValue;
end;

procedure TfrmReports.FormCreate(Sender: TObject);
begin
  InitFormState(Self);
  FUpdate := True;
  GetMonthNames(eMonth.Items); //наполняем контрол месяца значениями
  eMonth.ItemIndex := RestoreParam(Self, 'Month', MonthOf(Now) - 1); //читаем предыдущие пользовательские
настройки периода из реестра
  eYear.Value := RestoreParam(Self, 'Year', YearOf(Now));
  dReportEmp.ReadOnly := not DM.FIsAdmin; //запрещаем редактировать исполнителя не администратору
  FUpdate := False;
  RefreshData;
end;

procedure TfrmReports.FormDestroy(Sender: TObject);
begin
  StoreParam(Self, 'Month', eMonth.ItemIndex); //сохраняем пользовательский выбор периода в реестр
  StoreParam(Self, 'Year', eYear.IntValue);
end;

function TfrmReports.GetCurDate: TDate;
begin
  Result := EncodeDate(eYear.IntValue, eMonth.ItemIndex + 1, 1);
end;

procedure TfrmReports.RefreshData;
begin
  dReport.Close;
  dReport.SetVariable(0, GetCurDate);
  dReport.SetVariable(1, IncMonth(GetCurDate, 1));
  dReport.Open;
end;
end.

```