

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки, специальности)

Бизнес-информатика
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка элементов CRM-системы ООО «ИК «Сибинтек»

Студент

Д.В. Шляхов

(И.О. Фамилия)

_____ (личная подпись)

Руководитель

А.П. Тонких

(ученая степень, звание, И.О. Фамилия)

Тольятти 2020

Аннотация

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА, БАЗА ДАННЫХ, МОДЕЛЬ, CRM-СИСТЕМА, ПРОЕКТИРОВАНИЕ, АВТОМАТИЗАЦИЯ, УЧЕТ ЗАЯВОК, ПРАВИЛА МАРШРУТИЗАЦИИ ВХОДЯЩИХ ЗАЯВОК, СВОДНЫЕ ОТЧЕТЫ.

В данной бакалаврской работе рассматривается метод повышения эффективности процесса учета и управления входящими заявками в ООО «ИК «Сибинтек», который состоит в проектировании, разработки и внедрении новой информационной системы автоматизации обработки входящих заявок: АИС «Сибинтек: Управление заявками».

В практической части работы приведены основные модели спроектированного программного обеспечения системы: модель БД, архитектурные модели компонентов системы. Также выполнена реализация программного и информационного обеспечения системы по построенным моделям.

В заключительной части проведено тестирование разработанной системы и подтверждена ее функциональная корректность.

Работа представлена на 73 листах, рисунков 36, таблиц 5, приложений 3.

Оглавление

| | |
|---|----|
| Введение | 5 |
| Глава 1 Функциональное моделирование процесса управления заявками клиентов в ООО «ИК «Сибинтек» | 7 |
| 1.1 Технико-экономическая характеристика ООО «ИК «Сибинтек» | 7 |
| 1.2 Концептуальное моделирование процесса управления заявками клиентов в ООО «ИК «Сибинтек» | 10 |
| 1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям | 22 |
| 1.4 Постановка задачи на разработку проекта создания АИС управления заявками клиентов – АИС «Сибинтек: Управление заявками» | 25 |
| Глава 2 Логическое проектирование АИС «Сибинтек: Управление заявками» | 29 |
| 2.1 Описание технологии логического моделирования системы | 29 |
| 2.2 Логическая модель АИС «Сибинтек: Управление заявками» и ее описание..... | 30 |
| 2.3 Информационное обеспечение АИС «Сибинтек: Управление заявками» | 32 |
| 2.4 Проектирование базы данных АИС «Сибинтек: Управление заявками» | 35 |
| 2.5 Требования к аппаратно-программному обеспечению АИС «Сибинтек: Управление заявками» | 40 |
| Глава 3 Физическое проектирование АИС «Сибинтек: Управление заявками» | 43 |
| 3.1 Описание архитектуры АИС «Сибинтек: Управление заявками» | 43 |
| 3.2 Выбор технологии разработки программного обеспечения АИС «Сибинтек: Управление заявками» | 45 |
| 3.3 Выбор СУБД АИС «Сибинтек: Управление заявками» | 48 |
| 3.4 Реализация физической модели базы данных АИС «Сибинтек: Управление заявками» | 50 |

| | |
|--|----|
| 3.5 Разработка программного обеспечения АИС «Сибинтек: Управление заявками» | 53 |
| 3.6 Тестирование программного проекта | 62 |
| Заключение | 70 |
| Список используемой литературы | 71 |
| Приложение А Скрипт формирования физической структуры базы данных АИС «Сибинтек: Управление заявками» на языке описания данных DDL (Data Description Language) | 74 |
| Приложение Б Чертежи форм документов и видеокадров АИС «Сибинтек: Управление заявками» | 78 |
| Приложение В Фрагменты исходного кода программных модулей АИС «Сибинтек: Управление заявками» на языке программирования высокого уровня | 95 |

Введение

CRM (Customer Relationship Management) системы – прикладное программное обеспечение для компаний, предназначенное для автоматизации стратегий взаимодействия с клиентами [1].

С 2017 года на рынке программных решений по автоматизации работы с клиентами образовалась положительная динамика: множество компаний, стремясь оптимизировать расходы, стали внедрять инструменты, позволяющие повысить продуктивность процессов взаимодействия с клиентами, партнерами и поставщиками, вследствие чего возрос также спрос на CRM-системы [19].

По оценкам различных экспертов внедрение CRM-системы дает в среднем рост продаж на 20% [14].

Внедрение CRM-системы имеет следующие преимущества:

- наработка стандартизированной базы контактов;
- эффективный контроль качества работы отдела продаж;
- получение статистики и аналитики эффективности работы с входящими звонками, запросами;
- планирование стратегии развития бизнеса.

Объектом исследования в данной работе выступает компания ООО «ИК «Сибинтек».

Предметом исследования в данной работе является разработка системы автоматизации обработки заявок клиентов (CRM-системы) – АИС «Сибинтек: Управление заявками».

Цель данной работы – провести исследование предметной области, сформулировать требования для CRM-системы и разработать систему в соответствии с требованиями.

Для достижения поставленной цели в работе ставятся следующие задачи:

- разработать функциональное описание предметной области, выявить основные недостатки и предложить их устранение за счет применения CRM-системы;
- описать прецеденты предметной области и разработать требования к системе;
- разработать проект и базу данных для системы;
- выполнить реализацию CRM-системы в соответствии с поставленными требованиями.

Структура настоящей пояснительной записки включает в себя три главы и приложения.

Первая глава – аналитическая. В ней проводится обзор объекта исследования, дается его общая технико-экономическая характеристика, выбирается комплекс задач, необходимых для автоматизации. На основе поставленных задач определяются основные требования к информационной системе, проводится обзор аналогов рынка программного обеспечения.

Во второй главе осуществляется построение логической модели системы, описываются ее прецеденты, приводится процесс проектирования информационного обеспечения системы и базы данных.

В третьей главе приводится структура и архитектура системы, разрабатывается программное обеспечение. Все концептуальные решения в данной главе сопровождаются иллюстрациями соответствующих моделей на языке визуального моделирования UML.

В приложении А приведен листинг исходного кода формирования структуры БД на языке описания данных (DDL) для СУБД MySQL, полученный системой Enterprise Architect в автоматизированном режиме.

В приложении Б приведены примеры форм пользовательского интерфейса АИС «Сибинтек: Управление заявками».

В приложении В приведены фрагменты программного кода наиболее важных модулей программного обеспечения АИС «Сибинтек: Управление заявками» на языке программирования C#.

Глава 1 Функциональное моделирование процесса управления заявками клиентов в ООО «ИК «Сибинтек»

1.1 Техничко-экономическая характеристика ООО «ИК «Сибинтек»

ООО «ИК Сибинтек» работает на рынке ИТ-сервиса и аутсорсинга с 1999 года и является одним из лидеров отрасли. В портфеле компетенций ООО «ИК Сибинтек» представлен широкий спектр услуг сервисного обслуживания предприятий, системной интеграции, автоматизации технологических процессов, разработки и внедрения информационных систем, бизнес-консалтинга и сервисной интеграции, проектной деятельности и бизнес-приложений, комплексного создания ИТ-инфраструктуры. Разветвленная филиальная сеть и мощная инфраструктура ООО «ИК Сибинтек» позволяют компании реализовывать амбициозные проекты в сфере инноваций и информатизации бизнес-процессов для крупнейших корпораций, гарантируя индивидуальный подход к каждому клиенту и широкие возможности масштабирования внедряемых решений.

В ООО «ИК Сибинтек» работают квалифицированные профессионалы, обеспечивающие оказание сервисных услуг от Владивостока до Москвы. В таблице 1 приведены основные технико-экономические показатели компании ООО «ИК Сибинтек» по данным официального сайта [15] на 2020 год.

Таблица 1 – Техничко-экономические показатели ООО «ИК Сибинтек»

| Показатель | Значение |
|-----------------------------|--------------|
| Обслуживаемых пользователей | свыше 100000 |
| Информационных систем | свыше 500 |
| Предоставляемых сервисов | свыше 2000 |
| Заказчиков | свыше 100 |
| Реализуемых проектов в год | свыше 500 |

Как видно из таблицы 1, компания ООО «ИК «Сибинтек» (далее Компания) является утвердившей свои позиции и востребованной на рынке услуг ИТ-аутсорсинга, является развитой достаточно успешной.

Деятельность компании организована благодаря слаженной работе многочисленных специалистов. Схема организационной структуры управления ООО «ИК Сибинтек» приведена на рисунке 1.

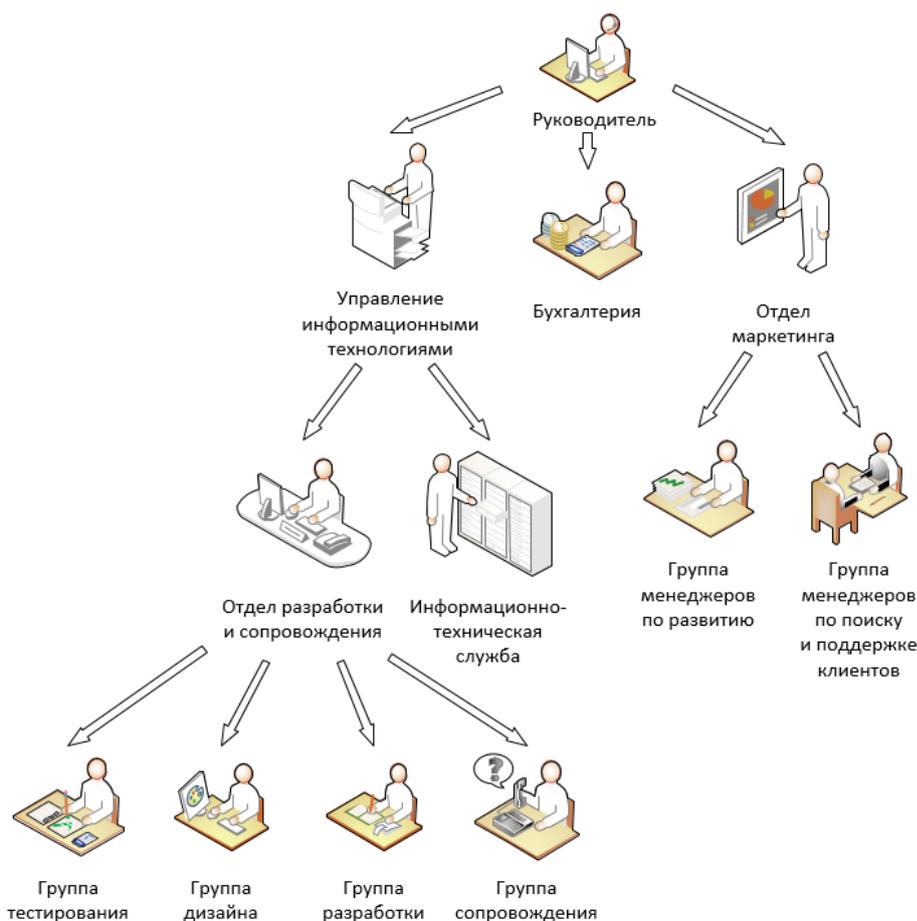


Рисунок 1 – Организационная структура Компании

Организационная структура Компании включает в себя следующие подразделения:

- главный руководитель – директор Компании – осуществляет общее управление Компанией;

- управление информационными технологиями, включающее в свой состав два подразделения: отдел разработки и сопровождения и информационно-техническую службу;

- бухгалтерия, которая занимается общими финансовыми вопросами Компании;

- отдел маркетинга, задачами которого являются перспективное планирование деятельности Компании и организация взаимодействия с клиентами.

Информационно-техническая служба является встроенным ИТ-сервисом Компании, в задачи которого входят обслуживание технической и программной инфраструктуры Компании.

Отдел разработки и сопровождения реализует основную деятельность Компании по разработке, внедрению, обслуживанию программно-технических решений, в том числе и собственного производства.

В рамках отдела разработки и сопровождения функционируют следующие структуры:

- группа разработки – разработка программно-технических решений;
- группа дизайна – разработка дизайна и интерфейсов для программно-технических решений;

- группа тестирования – оценка качества разработанных программно-технических решений;

- группа сопровождения – внедрение и обслуживание разработанных программно-технических решений.

Таким образом, организационная схема Компании имеет вид линейной иерархической структуры, уровень иерархии которой не превышает четырех ступеней. Преимуществами таких структур являются [11]: четко-определенная система взаимосвязей, согласованность, оперативность принятия решений.

К недостаткам линейных организационных структур можно отнести малую гибкость и приспособляемость, наличие большого количества

управляющих узлов (для крупных компаний) и общую длину связей между высшим руководителем и работником.

1.2 Концептуальное моделирование процесса управления заявками клиентов в ООО «ИК «Сибинтек»

Для демонстрации бизнес-процессов, анализа их архитектуры в целом и принятия решений об оптимизации имеются специальные методики и языки моделирования [16].

Методология SADT использует структурный подход к моделированию посредством структурного анализа и проектирования и базируется на графическом представлении систем в виде системы функций. Для существующих систем нотации IDEF методологии SADT могут быть использованы для анализа функций, выполняемых системой, а также для указания механизмов, посредством которых они осуществляются [6].

Модель SADT представляет собой серию диаграмм с сопроводительной документацией, разбивающих сложный объект на составные части, которые представлены в виде блоков. Каждая детальная диаграмма является декомпозицией блока из более общей диаграммы. На каждом шаге декомпозиции более общая диаграмма называется родительской для более детальной диаграммы [6].

В рамках SADT используется несколько нотаций моделирования. Основная из них – IDEF0, ее концептуальный принцип состоит в представлении любой анализируемой системы в виде набора взаимосвязанных блоков [6].

IDEF0 сводится к декомпозиции основных функций на отдельные бизнес-процессы, работы и действия, в результате чего разрабатывается иерархическая модель; при этом декомпозицию можно производить многократно, до четкого и детального описания всех процессов [6].

Одной из технологий описания бизнес-процессов, относящихся к типу «work-flow» – моделей, является нотация моделирования BPMN (Business Process Model Notation) – технология моделирования и нотации бизнес-процессов [3], представляющая средства для отражения подробных и важных понятий моделируемого процесса. Благодаря абстрактному представлению модели нотация BPMN позволяет наглядным образом описывать модели бизнес-процессов независимо от среды их функционирования. Полученная в результате модель представляет собой граф объектов предметной области, которые изображают действия и события, связанные между собой потоками управления.

Методология EPC (Event-Driven Process) является расширением нотации IDEF3, дополняя ее таким понятием события. Диаграммы EPC представляют собой упорядоченный граф событий бизнес-функций. Нотация EPC предлагает простое, наглядное и эффективное средство моделирования, позволяющее в виде последовательности событий и функций описывать сложные бизнес-процессы. Она применяется в таких распространенных программных продуктах, как SAP и ARIS [23].

Методология ARIS применяется при комплексном описании деятельности предприятия. Данная технология позволяет документировать бизнес-процессы, определить их особенности и недостатки, проанализировать эффективность работы предприятия, получить на основе моделей бизнес-процессов нормативные документы для поддержки и автоматизации.

В настоящей работе будет проводиться функциональное моделирование предметной области с декомпозицией функций с последующей идентификацией тех функций, которые будут автоматизированы. Для этих целей лучше всего из предложенных методологий описания бизнес-процессов подойдет методология структурного анализа и моделирования SADT.

В работе будут использованы нотации структурной модели IDEF0, диаграммы потоков данных DFD, цепочки событий IDEF3.

На рисунке 2 приведена контекстная диаграмма бизнес-процесса оказания услуг и взаимодействия с клиентами в ООО «ИК «Сибинтек». Диаграмма составлена с точки зрения системного аналитика с целью идентификации подпроцессов, которые оказывают негативное влияние на общую эффективность и которые можно автоматизировать.

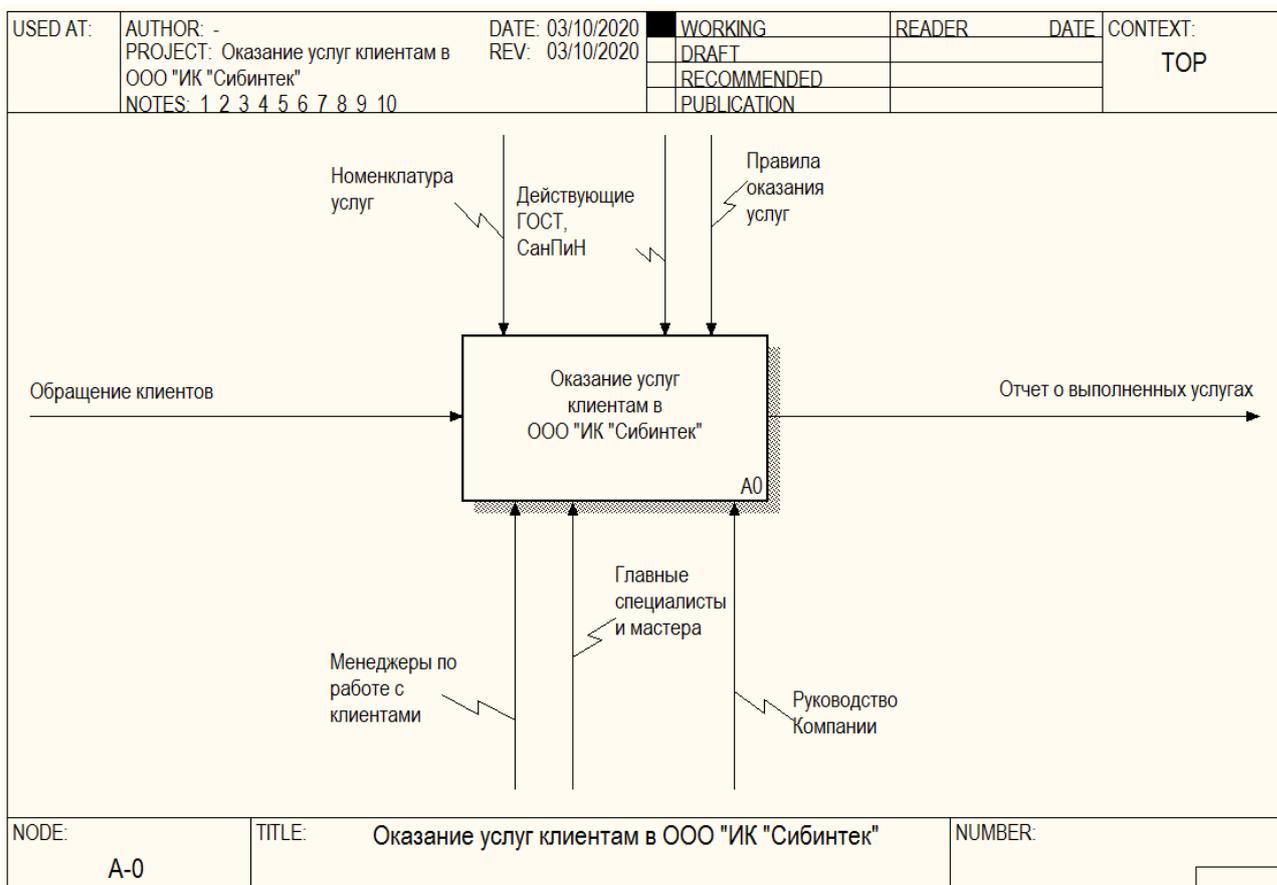


Рисунок 2 – Контекстная диаграмма процессов в ООО «ИК «Сибинтек»

Входящими данными для процесса оказания услуг клиентам в ООО «ИК «Сибинтек» являются обращения клиентов.

Результат этого процесса есть статистика, которая komponуется в отчет об оказанных услугах.

Непосредственными участниками, задействованными в моделируемом бизнес-процессе, являются менеджеры, обрабатывающие заявки клиентов, главные специалисты (руководители проектов) и мастера-исполнители (ведущие и сопровождающие проекты), а также руководство Компании. При всем этом перечисленные участники процесса руководствуются государственными стандартами и действующими СанПиНами в области оказываемых услуг, а также правилами оказания услуг, прописанными в действующем уставе Компании.

На рисунке 3 приведена диаграмма функциональной декомпозиции моделируемого процесса.

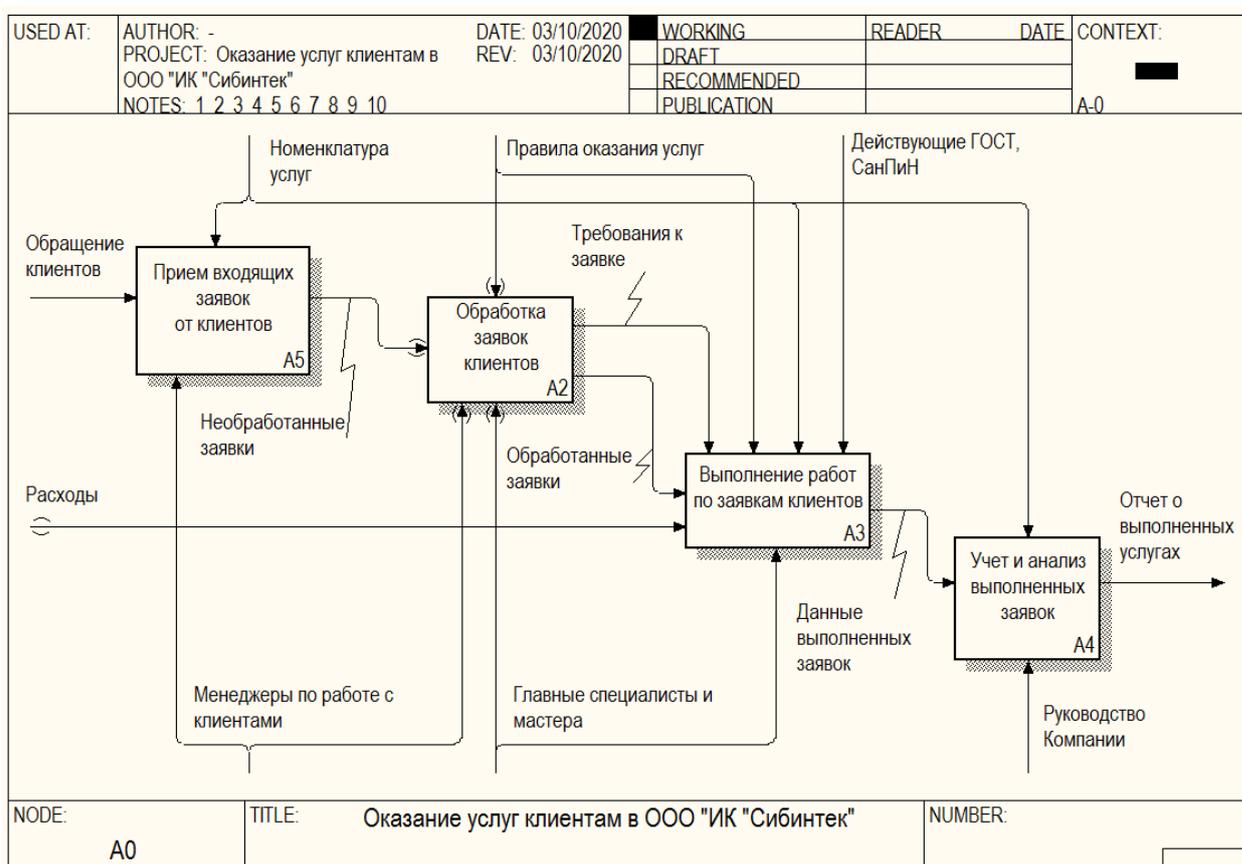


Рисунок 3 – Диаграмма декомпозиции процессов в ООО «ИК «Сибинтек»

Исходя из диаграммы рис. 3, взаимодействие с клиентами в ООО «ИК «Сибинтек» осуществляется следующим образом:

- прием входящих заявок от клиентов, в результате которого накапливаются «сырые» заявки, требующие обработки компетентными исполнителями;
- принятые «сырые» заявки обрабатываются менеджерами по виду запрошенных услуг, в результате чего эти заявки делегируются соответствующим компетентным исполнителям;
- непосредственное обслуживание (выполнение работ) заявок клиентов компетентными исполнителями;
- ведение учета выполненных заявок, составление сводного отчета о выполнении заявок для руководителя Компании.

Чтобы выделить функции, которые подлежат автоматизации, необходимо провести декомпозицию и более детально рассмотреть каждую из задач.

На рисунке 4 приведена диаграмма декомпозиции задачи приема заявок от клиентов. Данная диаграмма выполнена с помощью средств нотации IDEF3. На диаграмме показан порядок приема заявки клиента и средства, которые для этого применяются.

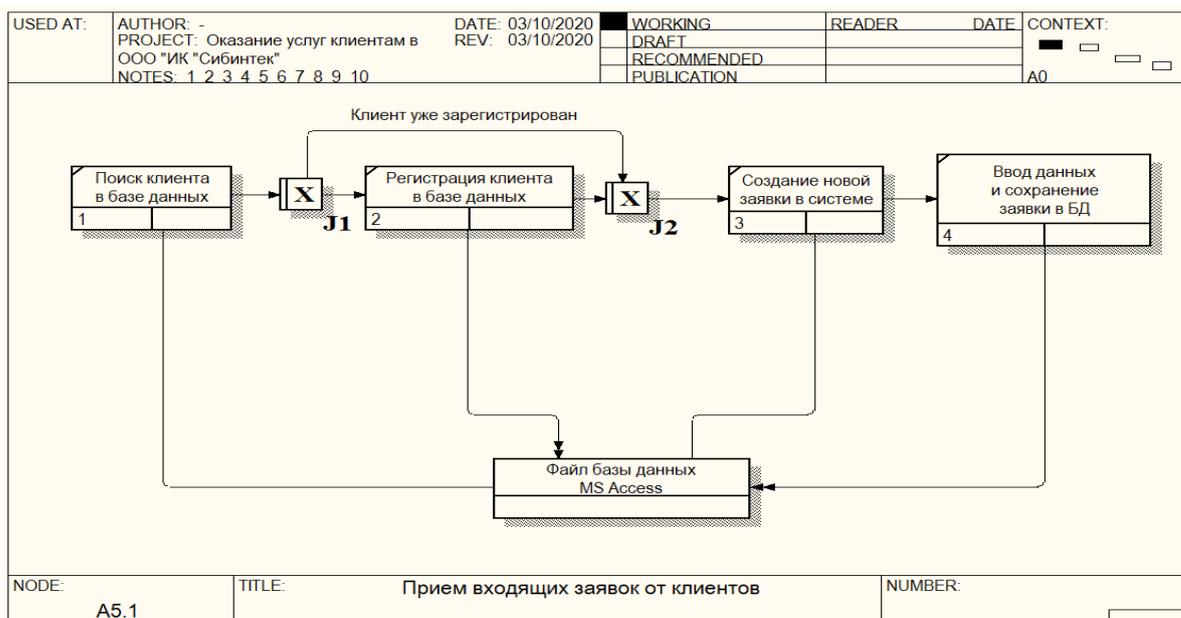


Рисунок 4 – Диаграмма декомпозиции задачи приема заявок

Так, при обращении клиента менеджер выполняет поиск клиента в существующей базе Компании (база данных клиентов в компании выполнена в виде файла .mdb, который является локальной базой данных СУБД MS Access).

Если клиента еще нет в базе данных компании (например, клиент обращается впервые), то менеджер создает новую запись о клиента.

Затем менеджер вводит в систему данные по заявке клиента с указанием типа заявки и всех необходимых данных со слов клиента. Заявка сохраняется в базе данных заявок. База данных заявок представляет собой отдельный файл, размещенный в папке общего доступа на сетевом хранилище. Файл базы данных заявок также выполнен в виде файл-серверной СУБД MS Access.

На рисунке 5 представлена диаграмма декомпозиции процесса обработки заявок. Для построения этой диаграммы выбрано представление нотации DFD-диаграмм потоков данных.

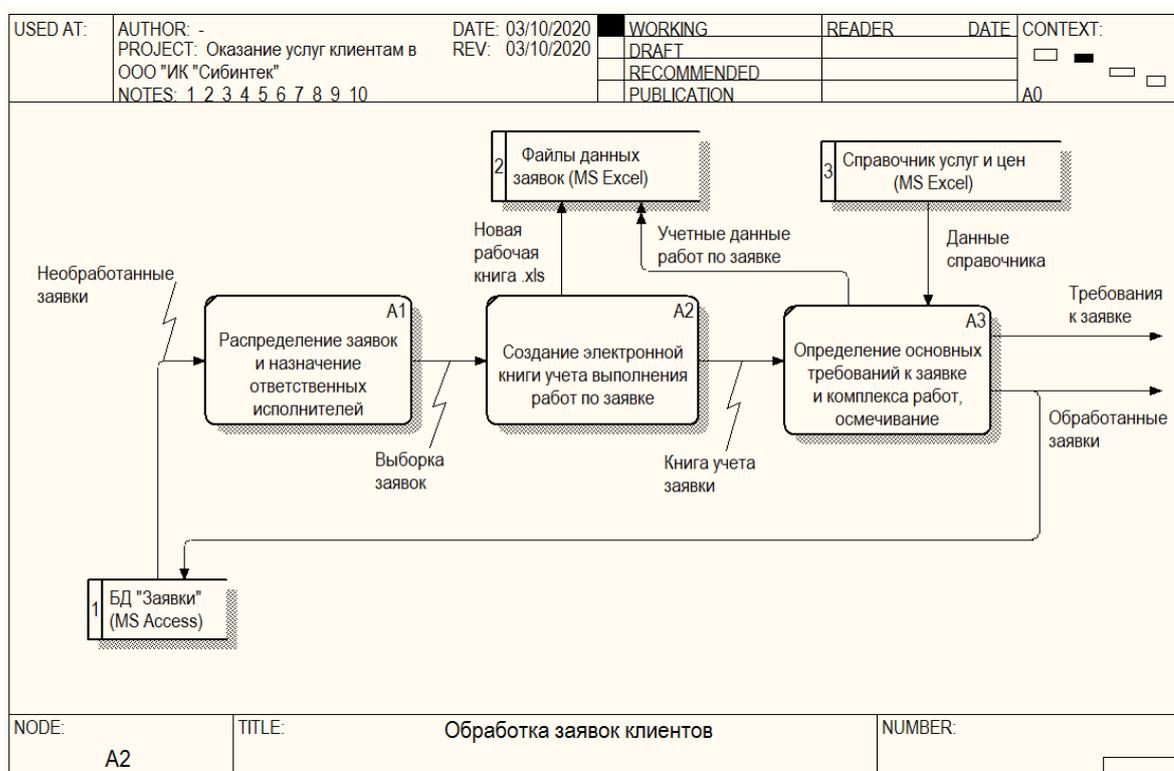


Рисунок 5 – Диаграмма декомпозиции задачи обработки «сырых» заявок

Как видно из диаграммы, «сырые» заявки группируются по типу запрошенных услуг (для исполнителей с различными компетенциями). Затем по каждой заявке создается ее учетная электронная книга (за основу берется электронная книга офисного приложения MS Excel). Все учетные книги по заявкам также лежат в папках общего доступа в разделе ответственного исполнителя. Данный файл наполняется в процессе выполнения заявки – в него вносятся сроки выполнения различных работ, затраты и т.д.

Руководитель подразделения, ответственного за выполнение заявки, рассчитывает смету предварительной стоимости заявки, руководствуясь объемами работ, запрошенными клиентом, а также справочником цен на услуги компании (справочник цен на услуги существует в компании в виде файла MS Excel, размещенного в общем доступе на сервере общих папок).

После всех подготовительных действий и оформления соответствующих документов начинается выполнение работ по заявкам. Диаграмма данного процесса в нотации IDEF3 приведена на рисунке 6.

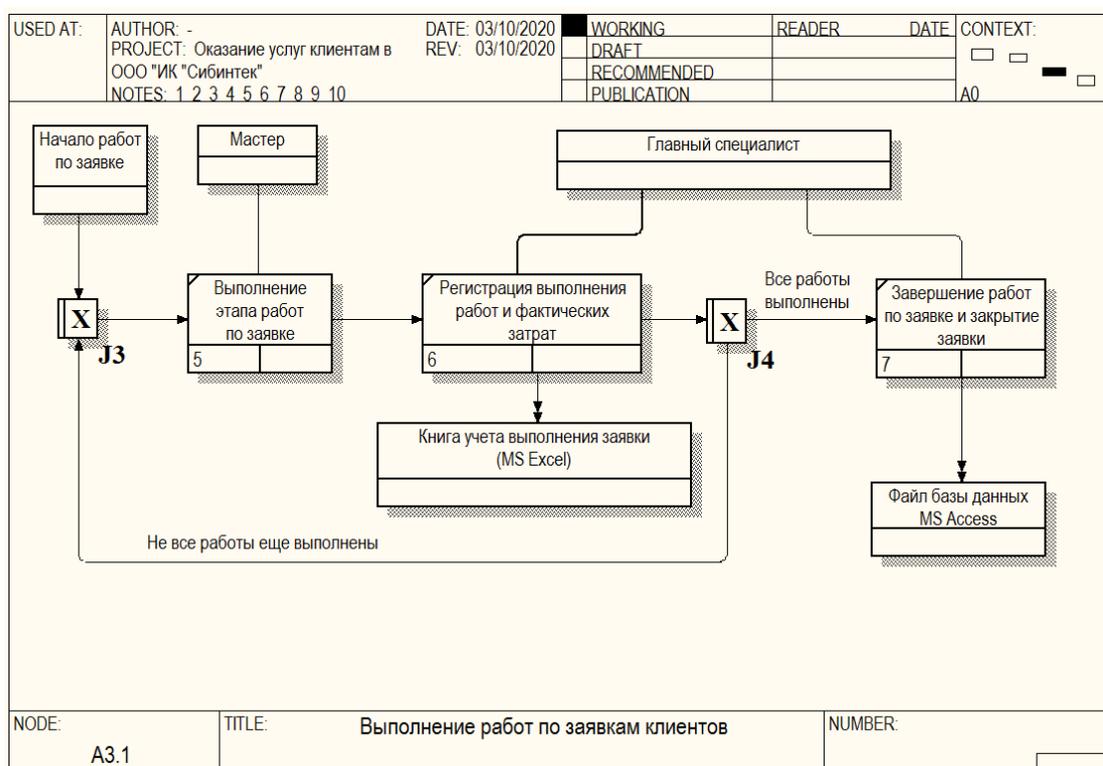


Рисунок 6 – Диаграмма декомпозиции выполнения заявок

Выполнение заявки представляется выполнением всех запланированных работ с внесением соответствующих записей в подготовленную электронную рабочую книгу заявки. После выполнения заявки в файл MS Access в запись о заявке вносятся сведения о ее решении.

При наступлении отчетного периода руководитель компании анализирует проделанную работу. Для этого он запрашивает у руководителей подразделений отчеты по выполненным объемам работ за отчетный период. Для этого руководители подразделений просматривают и консолидируют данные из файлов базы данных заявок и файлов MS Excel с данными о выполнении каждой из заявок, попадающих в отчет.

Так создаются сводные отчеты, которые отправляются руководителю лично или через корпоративные каналы связи.

На рисунке 7 приведена IDEF3-диаграмма, показывающая процесс консолидации данных и составления сводного отчета.

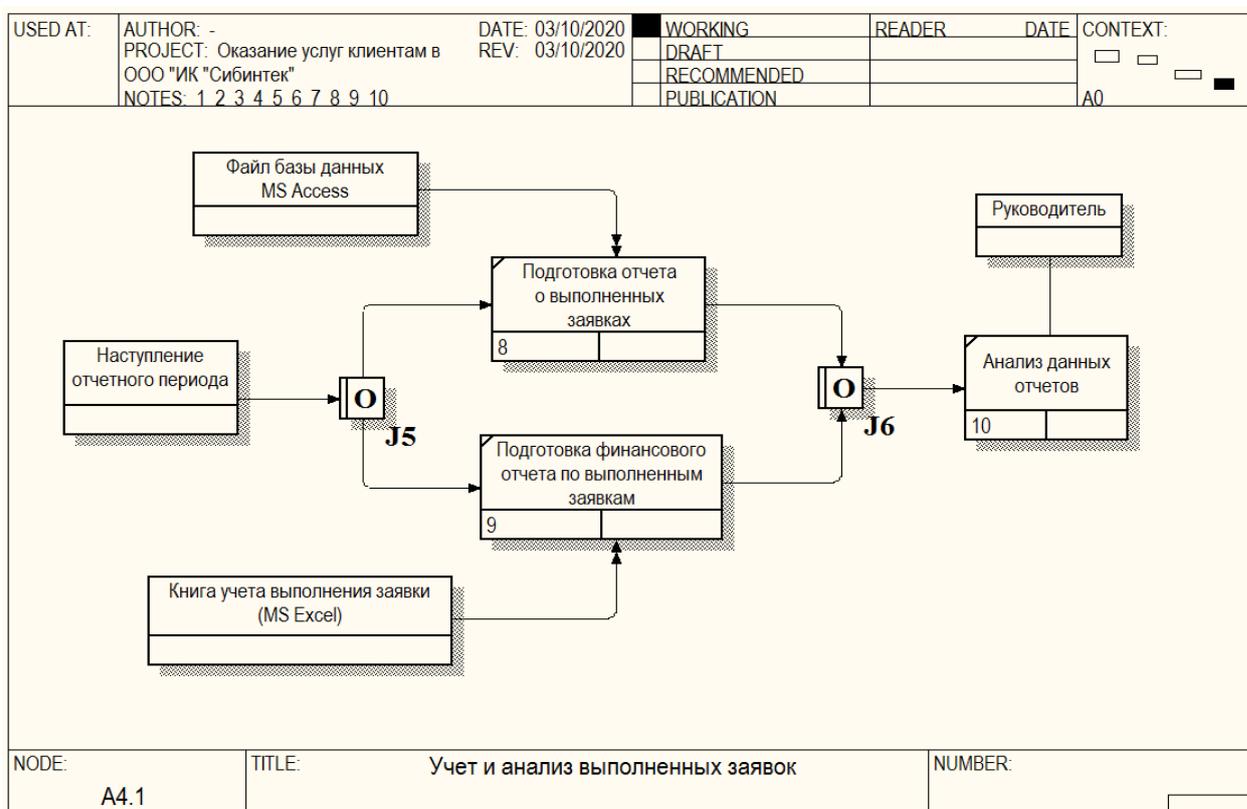


Рисунок 7 – Диаграмма декомпозиции задачи подготовки сводного отчета

Таким образом, из проведенного анализа задач компании по обработке заявок видно, что программно-инструментальной основой процесса является совместное использование офисных приложений MS Access и MS Excel.

Использование офисных приложений MS Access и MS Excel в качестве программно-инструментальной основой процессов обработки заявок является неэффективным и оказывает негативное влияние на такие характеристики информационных процессов, как: скорость, надежность, достоверность.

На рисунке 8 показана типовая схема жизненного цикла заявки, построенная в соответствии с приведенным ранее описанием.

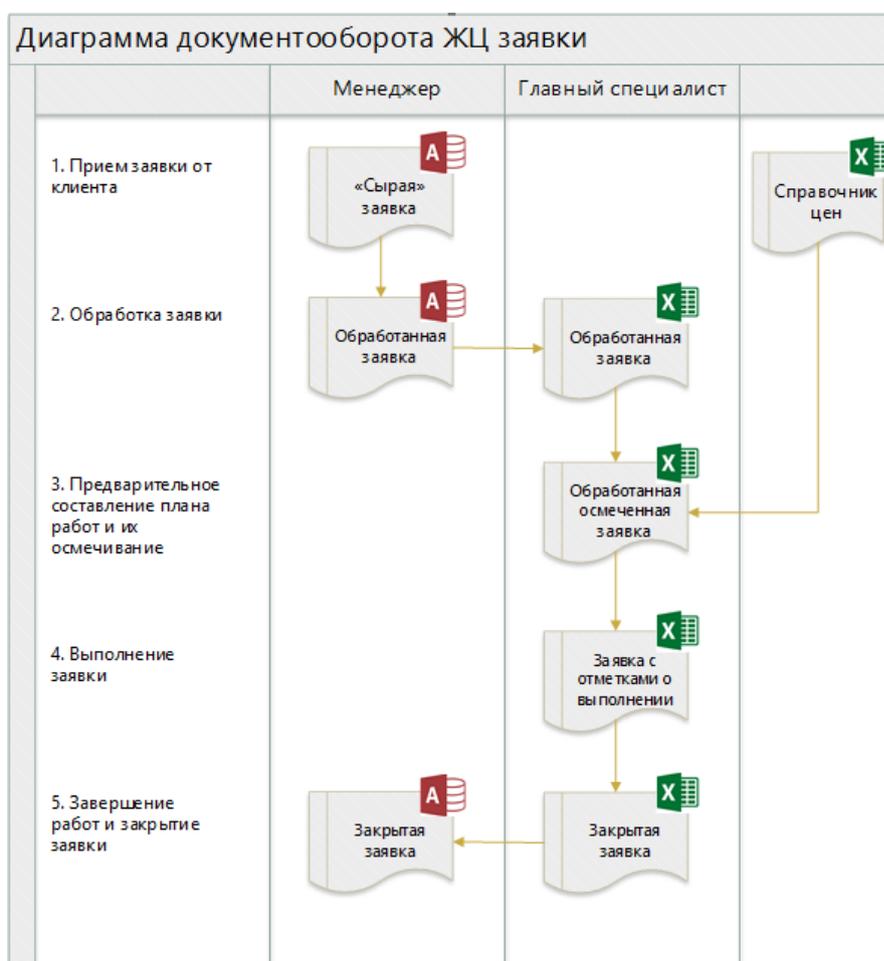


Рисунок 8 – Схема жизненного цикла заявки

Сначала заявка регистрируется в системе: сама заявка и клиент,

который обратился. При этом, как уже было сказано, использовать СУБД MS Access неэффективно.

Далее «сырая» заявка поступает на обработку, где менеджер должен определить вид заявки и делегировать ее соответствующему компетентному исполнителю. Здесь можно увидеть у менеджера «лишнюю» функцию, поскольку функция поиска ответственного исполнителя в зависимости от вида заявки вполне может быть автоматизирована, так как номенклатура исполнителей и компетенций – конечное множество, а для каждого исполнителя заранее известны его компетенции. Так, новая заявка на основе правил маршрутизации может быть автоматически делегирована нужному исполнителю.

Ответственный исполнитель подготавливает смету на работы по заявке. При этом он руководствуется, как минимум, двумя книгами MS Excel (план заявки, справочник цен) и файлом MS Access (база заявок). Такая коллаборация программ и источников данных для выполнения сравнительно простой задачи неэффективна.

После выполнения и регистрации всех работ исполнитель закрывает заявку. При этом менеджеру и исполнителю требуется согласовать свои действия: менеджер должен закрыть заявку в БД (MS Access), а исполнитель – в своей учетной электронной книге. Так, менеджер для выполнения операции использует базу данных, а исполнитель – книгу учета выполнения заявок. Такая схема использования двух разных систем для решения одной задачи также неэффективна и ненадежна.

Далее, если рассмотреть схему подготовки сводного отчета по выполненным работам, то также можно выявить недостатки использования выбранных программных средств.

Как и везде, в процессе подготовки отчета также видно совместное использование офисных приложений MS Access и MS Excel. При этом, если в MS Access запросить выборку данных по заявкам в принципе не составляет труда (через мастера или конструктор запросов, обладая базовыми навыками

написания запросов на языке SQL), то в таблицах MS Excel это сделать очень трудно, поскольку за отчетный период может накопиться много заявок и все они разнесены по разным файлам. Для этого потребуется открывать каждую электронную книгу учета заявки, выбирать из нее сводные данные и сохранять их в каком-то промежуточном месте (например, в отдельной книге), а затем все эти данные еще раз сводить. Очевидно, что такая схема неоправданно отнимает большое количество времени и требует большой концентрации и внимания, а рутинная работа в больших объемах способна вызывать произвольное допущение ошибок у человека.

На рисунке 9 приведена схема документооборота процесса подготовки сводного отчета.



Рисунок 9 – Схема составления сводного отчета

Таким образом, подтверждается, что совместное использование офисных приложений MS Access и MS Excel неэффективно.

Рассмотренные схемы типовых и ключевых процессов компании позволяют сделать следующие заключения:

- выбранные программно-инструментальные средства неэффективны, так как требуют времени, не сопоставимого с фактической сложностью задач, негативно влияют на целостность данных и надежность их хранения, требуют максимальную концентрацию и согласованность, затрудняют элементарный поиск информации;

- менеджер имеет функции, которые можно автоматизировать;

- существующая система никак не контролирует безопасность и целостность, а также не обеспечивает надежность данных;

- все источники данных сильно децентрализованы, что снижает эффективность работы с ними;

- клиенты не получают обратной связи о состоянии своих заявок.

Если внедрить автоматизированную информационную систему, обеспечивающую:

- централизованное хранение, ввод и обработку данных (при этом СУБД будет заменена на клиент-серверную).

- контроль вводимых данных, обеспечение целостности данных и согласованность при выполнении операций;

- автоматизированное ведение справочника услуг;

- автоматизированное назначение ответственных исполнителей на заявки (обеспечиваемое за счет введения правил маршрутизации входящих заявок);

- автоматизированное назначение сроков выполнения заявок в зависимости от их типа и других характеристик;

- автоматизированное ведение учета выполнения заявок;

- автоматизированное составление наглядных сводных отчетов,

то недостатки использования старого стека программно-инструментальных средств можно устранить.

1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям

Множество направлений бизнес-процессов, направленных на учет заявок клиентов (так называемых, тикет-систем), требует от производителей систем предусматривать как можно больше особенностей каждого из процессов. Однако невозможно учесть все возможные требования для заказчика во всех частных случаях. Каждый разработчик в своей системе старается создать высокую степень универсальности продукта. Наиболее универсальной система может получиться, если реализовать в ней максимальный набор возможных функций. Возрастание числа функций также увеличивает затраты на ее разработку. Поэтому, чем больше функций предлагает система, тем дороже она будет являться для заказчика. Кроме того, зачастую в таких системах множество функций поверхностны и не продуманы.

Одним из примеров тикет-систем является продукт, организованный на популярной платформе 1С – «1С: Предприятие», в конфигурации «Учет заявок». Как и любой другой продукт от 1С, «1С: Предприятие» – это полностью конфигурируемая система, которая должна быть настроена под конкретное предприятие. Применение продуктов 1С, как правило, требует аутсорсинга для обслуживания или наличия в собственном штате компетентного 1С-программиста. В обоих случаях это влечет дополнительные периодические затраты.

Другим примером системы учета может являться программный продукт «Helpdeskdu». Подходит для ведения малого бизнеса, предоставляет функции управления заявками, доступа к базе знаний,

формирования отчетов, настройку аккаунтов пользователей, управление контактами. Программа распространяется на коммерческой основе.

В качестве альтернативных вариантов можно также рассмотреть варианты зарубежных Open Source тикет-систем.

Одной из таких систем является OTRS (Open-source Ticket System) – система, популярная в больших корпорациях (Sony, HP, Yandex). Данная система позволяет вносить существенные изменения в код и фактически создавать собственную тикет-систему. Основными недостатками этой системы являются ее высокая сложность в настройке, интеграции и обслуживании и, как следствие, наличие штатного специалиста для обслуживания системы. Данная система также имеет интерфейс, к которому требуется приспособиться. Таким образом, применение данной системы для поставленной задачи является неоправданно сложным.

Еще одной системой является Request Tracker – тикетная система с хорошо структурированным открытым кодом. Схожа с OTRS и в исходном состоянии предоставляется в «сыром» виде – поэтому потребует доработки сразу после установки. Очевидно, что недостатки данной системы аналогичны недостаткам системы OTRS – сложность в настройке и обслуживании, наличие штатного программиста для наладки системы.

Таблица 2 – Основные функциональные возможности продуктов

| Требование/ Характеристика | 1С: Предприятие. | Helpdeskddy | OTRS | Request Tracker |
|---------------------------------------|---|-------------------------------------|---|----------------------------------|
| Стоимость (на 30-50 рабочих мест) | 228 400 руб. (10 активных пользователей) | 20 евро за 1 пользователя в месяц | 30\$ за одного пользователя (бессрочно) | Бесплатно |
| Возможность настройки | При выборе конфигурации | Интеграция с ArmoCRM, YouScan, | Интеграция с Active Directory | Интеграция с RTFM |
| Возможность расширения функций | По прайсу 1С | Оговаривается отдельно с заказчиком | Оговаривается отдельно | Множество дополнительных модулей |

Продолжение таблицы 2

| Требование/ Характеристика | 1С: Предприятие. | Helpdeskddy | OTRS | Request Tracker |
|---------------------------------------|--------------------------------------|----------------------------|-------------------------------|-----------------------------|
| Возможность сопровождения | По прайсу 1С | 200000 руб. на 100 ч./мес. | В среднем 5000 рублей в месяц | 6000 руб./мес. |
| Обучение использованию | По прайсу 1С | Руководство пользователя | Руководство пользователя | Руководство, ресурсы в сети |
| Ведение клиентской базы | + | + | + | + |
| Маршрутизация заявок | - | - | - | - |
| Правила обработки заявок | - | + | - | + |
| Управление заявками | + | + | + | + |
| Разграничение прав доступа | + | + | + | + |
| Возможность расширения функций | Программирование на встроенном языке | Нет | Возможно | Возможно (PHP+MySQL) |

Рассмотренные системы имеют недостатки, не позволяющие применить их как готовое решение для автоматизации задач в ООО «ИК «Сибинтек»:

- высокая стоимость приобретения / сопровождения;
- необходимость сложной и дорогой настройки после приобретения;
- неполное соответствие важным предъявленным требованиям;
- наличие дополнительных функций, которые повышают стоимость системы, но в использовании которых нет необходимости.

Последний недостаток является определяющим – нет смысла приобретать систему и платить немалые деньги за большое количество функций, которые не востребованы. Создание такой информационной системы позволит повысить эффективность учета обращений клиентов и управления заявками, поскольку ее функционал позволит:

- автоматизировать часть ручного труда по управлению заявками;

- понизить риск допущения произвольных ошибок при вводе информации вручную (человеческий фактор);
- анализировать фактические показатели обработки заявок;
- оптимизировать процессы хранения и доступа к данным;
- оперативно и быстро составлять и искать необходимые документы.

Таким образом, разработка собственного программного решения информационной системы учета обращений клиентов актуальна для Компании.

1.4 Постановка задачи на разработку проекта создания АИС управления заявками клиентов – АИС «Сибинтек: Управление заявками»

Функциональные требования, предъявляемые к АИС «Сибинтек: Управление заявками» определяются в рамках настоящей работы:

- ведение справочников: приоритетов заявок, производителей применяемых материалов (оборудования, программного обеспечения), типов заявок, типовых выполняемых работ, статусов заявок, подразделений, обрабатывающих (исполняющих) заявки;
- ведение контактной базы клиентов;
- учет аккаунтов пользователей;
- управление обработкой заявок, включая поддержку правил маршрутизации заявок, поддержку правил нормативного времени обработки заявок (в зависимости от их типа, приоритета);
- учет выполнения заявок с возможностью комментирования, управление статусами, регистрации выполненных работ, расчет полной стоимости заявки;
- формирование сводных отчетов: финансовый отчет по решенным (или выполняемым) заявкам, тренд, иллюстрирующий динамику обработки и

решения заявок, набор графических отчетов (инфографика) о решении выполнению заявок (по типу, итоговой цене, ответственным, статусам).

Нефункциональные требования определяют внутренние и внешние атрибуты функционирования системы.

Удобство использования продукта влияет на его привлекательность для пользователя, скорость и продуктивность процесса обучения пользователя с этим продуктом. Требования к удобству использования определяют такие характеристики программного продукта, как понятность, удобство работы, удобство обучения [18].

Требования к удобству использования, предъявляемые к АИС «Сибинтек: Управление заявками»:

- система должна иметь оконный интерфейс;
- все надписи, указания и пояснения должны быть выполнены на русском языке;
- компоновка интерфейса системы должна быть выполнена таким образом, чтобы все элементы были выстроены и логически сгруппированы, оснащены соответствующими заголовками или названиями;
- схожие компоненты интерфейса должны предоставлять или выполнять схожие функции;
- на уровне интерфейса необходимо максимально снизить возможность допущения ошибок ввода данных (например, для строго числовых данных использовать NumericUpDown – компоненты вместо TextBox, применять фиксированные списки там, где выбор ограничен и т.д.);
- система не должна требовать сложной настройки и должна быть готова к запуску сразу после установки, система должна быть портативной.

Требования к надежности предлагают повышать характеристики программного продукта, обеспечивающие такие характеристики, как зрелость, устойчивость, восстанавливаемость [17].

Требования к надежности, предъявляемые к АИС «Сибинтек: Управление заявками»:

- система должна работать под управлением ОС Windows при наличии соответствующих библиотек и расширений;
- в системе должна присутствовать диагностика ввода исходных данных, которая подразумевает контроль вводимых данных на соответствие типу, а также контроль обязательного заполнения данных;
- диалоговые формы ввода данных должны быть модальными;
- предусмотреть обработчик исключительных ситуаций, которые могут возникнуть в ходе эксплуатации, и реакцию на них.

Производительность программы определяет показатели, которые характеризуют эффективность использования системы по сравнению с ресурсами, которые выделяются для ее работы, и определяет такие черты продукта, как эффективность использования ресурсов, время использования ресурсов [18].

Требования к производительности, предъявляемые к АИС «Сибинтек: Управление заявками»:

- в случае если требуется обработка большого объема данных, необходимо предусмотреть соответствующую индикацию, которая будет сообщать о ходе выполнения операции (процесса);
- обеспечить своевременное обновление информации на дисплее в соответствии с действиями пользователя (максимальное время отклика – 5 секунд на выборку данных объемом 1000 строк).

Требования к удобству сопровождения определяют такие характеристики, как модифицируемость, стабильность, тестируемость [18].

Требования к сопровождению, предъявляемые к АИС «Сибинтек: Управление заявками»:

- система должна функционировать в операционной системе Windows с использованием платформы .NET Framework версии не ниже 4.5;
- разработка системы должна быть произведена в среде Visual Studio версии 2015-2017;

- архитектура ПО системы должна быть построена по модульному с вынесением функциональных групп модулей в отдельные пакеты;
- основные участки программного кода должны быть оснащены понятными комментариями на русском языке, а закрытые функции и «know-how» должны быть сопровождаемы подробной спецификацией.

Выводы по главе 1: в результате анализа бизнес-процессов обработки заявок клиентов в ООО «ИК «Сибинтек» была построена модель, идентифицированы основные недостатки процесса. На основании анализа были выделены те задачи, автоматизация которых позволит повысить общую эффективность рассмотренного процесса, были предложены конкретные способы решения.

Анализ предложений, актуальных на рынке программного обеспечения, показал, что существующие системы не смогут в полной мере обеспечить разработанные меры по автоматизации указанных процессов, либо будут избыточными в функциональном плане (при этом цена/качество не будет соответствовать ожиданиям).

В результате было предложено разработать новую систему АИС «Сибинтек: Управление заявками», которая обеспечит автоматизацию указанных задач. Для данной системы была разработана комплексная модель требований: функциональных и нефункциональных.

Глава 2 Логическое проектирование АИС «Сибинтек: Управление заявками»

2.1 Описание технологии логического моделирования системы

Логическая модель проектируемой АИС «Сибинтек: Управление заявками» должна отражать ее функциональность и показывать, как пользователь должен взаимодействовать с данной системой. Такая модель может быть выполнена в виде диаграммы прецедентов в нотации UML.

Диаграммы прецедентов предназначены для более удобной передачи информации между моделью системы и моделью разрабатываемого ПО [8]. С помощью диаграммы прецедентов можно смоделировать предметную область, позволив в будущем упростить процесс формализации функциональных требований. Преимущество построения логических моделей в виде диаграмм прецедентов состоит в высокой степени наглядности представления пользовательских требований к системе, а также детализации действующих лиц системы и отведенных им функций системы.

Само понятие прецедента раскрывает действия, которые может выполнять моделируемая система при выполнении пользователями соответствующих действий (совершаемых событий).

Так, диаграммы прецедентов могут представлять как пользовательские требования, так и логическую модель, описывающую предметную область моделируемой системы. Например, диаграмма прецедентов может быть источником для последующей идентификации основных классов и объектов предметной области. Выделение объектов системы, их атрибутов и связей между ними может быть основой при построении диаграммы классов, которая является основой объектно-ориентированного подхода к созданию системы [9].

Таким образом, в качестве описания логической модели проектируемой АИС «Сибинтек: Управление заявками» выбрана нотация UML-диаграмм.

2.2 Логическая модель АИС «Сибинтек: Управление заявками» и ее описание

Диаграмма прецедентов АИС «Сибинтек: Управление заявками» приведена на рисунке 10. Данная диаграмма выполнена с помощью CASE-системы Enterprise Architect версии 15.0 [12].

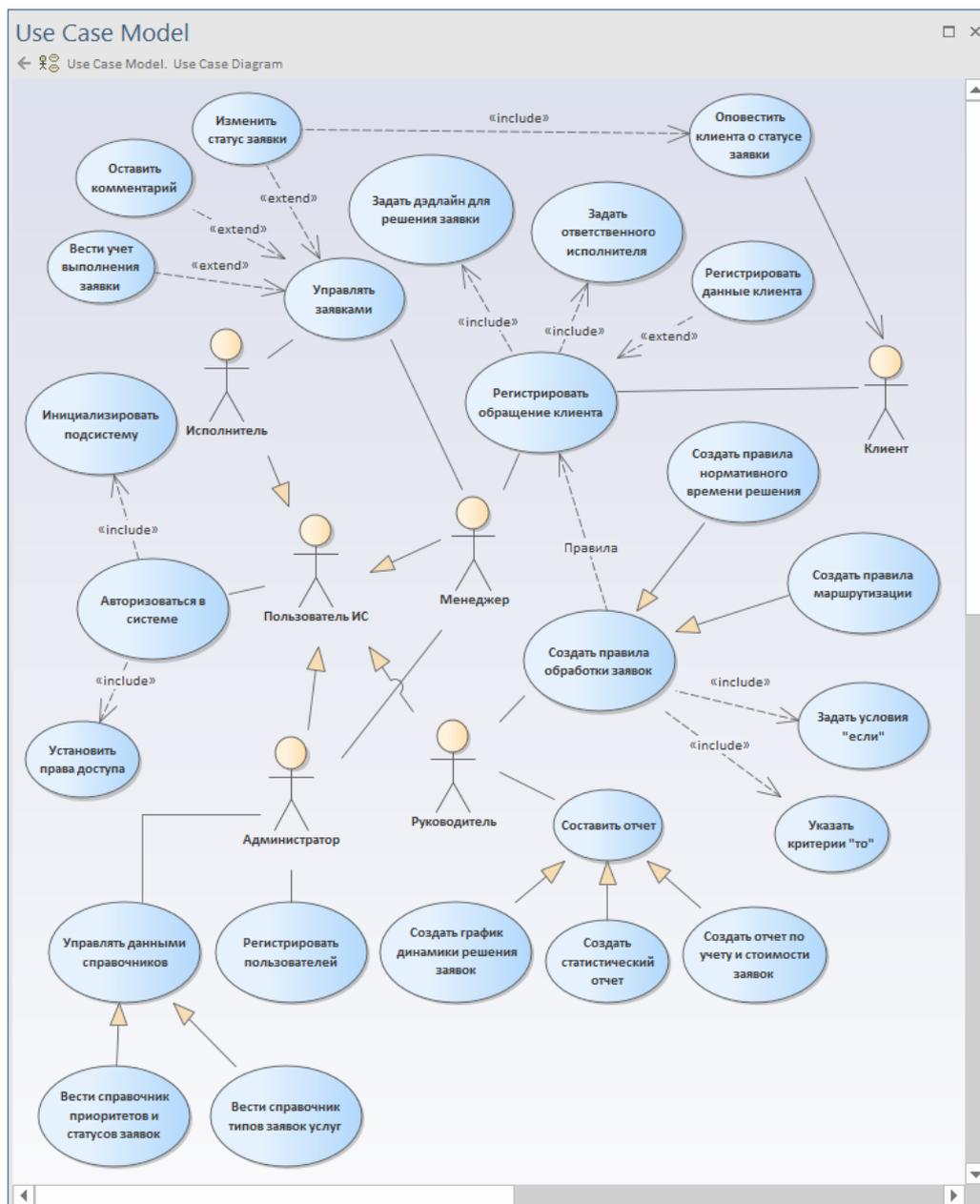


Рисунок 10 – Диаграмма прецедентов АИС

В таблице 3 приведены функции, отведенные пользователям системы в соответствии с построенной диаграммой прецедентов.

Таблица 3 – Функции пользователей АИС «Сибинтек: Управление заявками»

| Роль пользователя | Функции ИС |
|--------------------------|--|
| Администратор | <ul style="list-style-type: none"> – Регистрация пользователей ИС – Ведение справочника приоритетов заявок – Ведение справочника статусов заявок – Ведение справочника типов заявок – Ведение справочника услуг Компании – Создание правил маршрутизации заявок клиентов (автоматизированное назначение сроков и исполнителей в зависимости от типа и приоритета заявки, клиента) |
| Менеджер | <ul style="list-style-type: none"> – Регистрация клиентов и их данных – Регистрация заявок по обращениям клиентов и автоматическое назначение сроков и ответственных исполнителей на заявки (в соответствии с правилами маршрутизации) – Обзор заявок, построение сводной таблицы заявок – Оповещение клиентов о статусе их заявок |
| Исполнитель | <ul style="list-style-type: none"> – Автоматическое уведомление о новых заявках по профильному направлению исполнителя (в соответствии с правилами маршрутизации заявок) – Просмотр заявок, построение сводной таблицы заявок (только в которых определен как исполнитель) – Учет выполнения заявок – Изменение статуса заявки |
| Руководитель | <ul style="list-style-type: none"> – Обзор (учет, поиск) заявок, построение сводной таблицы заявок – Редактирование правил маршрутизации заявок клиентов – Построение финансового отчета по выполненным заявкам – Построение тренда, определяющего динамику выполнения заявок «создано-выполнено» – Построение графических отчетов по статистике обработки заявок клиентов в виде (группировка по типу заявок, по суммарной стоимости выполненных заявок, по ответственному исполнителю, количеству, статусу) |

2.3 Информационное обеспечение АИС «Сибинтек: Управление заявками»

Информационное обеспечение АИС «Сибинтек: Управление заявками» представляется набором таблиц, объединенных в реляционную базу данных. Информация в таблицы базы данных (БД) заносится в процессе работы АИС «Сибинтек: Управление заявками» вручную (при заполнении или редактировании справочных данных) и автоматически (при маршрутизации и назначении сроков выполнения заявок).

Вручную информация в базу данных заносится посредством специальных диалоговых форм, которые спроектированы с целью минимизировать количество ошибок ввода данных.

Информационные таблицы подразделяются на справочные (условно-постоянные) – данные в них предназначены для использования в других таблицах и редко подвергаются изменению, и оперативные, которые формируются в ходе функционирования системы с определенной частотой.

Перечень требуемых справочных таблиц составляют: подразделения, сотрудники, услуги, правила маршрутизации, типы, статусы и приоритеты обработки заявок. Перечень необходимых оперативных таблиц включает в себя данные заявок, выполненных работ, комментирования заявок.

Используемые в АИС «Сибинтек: Управление заявками» классификаторы предназначены для идентификации записей, организации учета справочников и обеспечения связывания информации в составе БД. Используемые классификаторы являются локальными в пределах системы, имеют линейную систему классификации с порядковым кодированием. Порядковые номера присваиваются автоматически при создании записей в БД посредством интегрированных возможностей СУБД в виде использования целочисленных полей-счетчиков с поддержкой автоинкремента.

Выходная информация представляется в АИС «Сибинтек: Управление заявками» в виде информационных таблиц учетных данных и документов.

Документы, которые формирует АИС «Сибинтек: Управление заявками», представляют собой информацию, полученную путем консолидации справочных и оперативных таблиц и приведенных к установленной форме выходного документа.

К результатной информации относятся:

- выводимые на формы справочные таблицы, к которым применяется фильтр поиска и сортировки – то есть обработанные данные справочников;
- отчеты по выполнению заявок – перечень выполненных работ и общий расчет стоимости заявок;
- общий отчет по выполненной работе сотрудниками Компании по всем заявкам за указанный период времени.

Управление всеми справочниками в АИС «Сибинтек: Управление заявками» имеет типовую информационную модель: выбор соответствующей таблицы, вызов формы ввода данных, ввод данных, сохранение изменений в справочнике (рисунок 11).



Рисунок 11 – Информационная модель ведения справочников

Приведенная модель справедлива для следующих справочников: отделы, сотрудники, услуги, правила обработки, типы заявок, статусы заявок, приоритеты обработки заявок.

Основное назначение АИС «Сибинтек: Управление заявками» – учет заявок при обращениях клиентов и ведение информации по обработке и выполнению заявок.

Информационная модель процесса обработки заявок в АИС «Сибинтек: Управление заявками» приведена на рисунке 12.

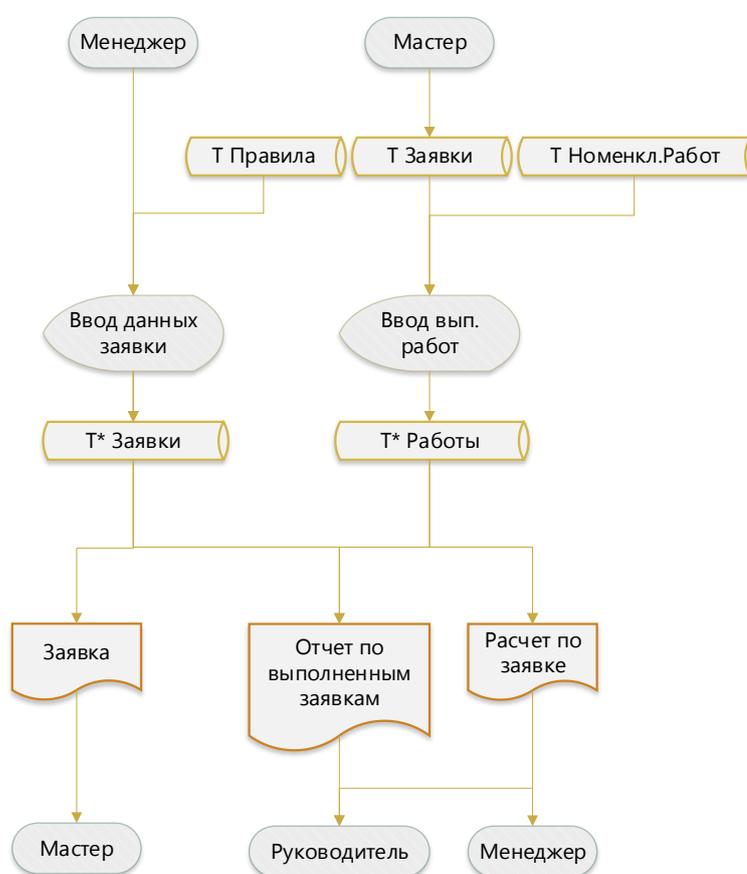


Рисунок 12 – Информационная модель обработки заявок в АИС «Сибинтек: Управление заявками»

Менеджер регистрирует поступающие заявки от клиентов в системе посредством специальной формы новой заявки. При вводе данных заявки используется справочник заданных правил обработки заявок. Руководствуясь

данными правилами, система определяет нормативные сроки обработки заявки и назначает ответственного исполнителя. После этого новая заявка сохраняется в базе данных. Данная заявка также направляется выбранному системой ответственному исполнителю.

По ходу выполнения заявок ответственные исполнители вносят в данные заявки информацию о произведенных работах.

Информация по выполненным работам по заявке позволяет сформировать отчет по стоимости заявки.

Учетная информация по всем заявкам системы позволяет сформировать итоговые отчеты по обслуживанию клиентов и выполненным заявкам. Такие отчеты составляются для руководителя для их последующего анализа.

2.4 Проектирование базы данных АИС «Сибинтек: Управление заявками»

Методология проектирования баз данных включает в себе структурированный подход, предусматривающий использование специализированных процедур, технических приемов, инструментов, документации и ориентированный на поддержку и упрощение процесса проектирования [4].

Процесс проектирования информационной базы данных начинается с детального описания и последующего анализа сущностей (объектов) предметной области [20]. Затем модель предметной области отображается на модель данных, совместимую с выбранной СУБД – физическую модель, определяющую схему размещения данных и методы доступа к ним. Модель физического уровня строится также с учетом возможностей, предоставленных выбранной СУБД.

Физическая модель данных получается после проработки вопроса об организации хранения данных в физических таблицах в БД. В ходе

разработки схемы данных составляется перечень информационных таблиц, с каждым полем этих таблиц сопоставляется конкретный домен (тип данных), поддерживаемый выбранной СУБД.

На модели также указывается обязательность (требования к заполненности) полей таблиц и их ключевые характеристики (идентифицирующие признаки и внешние ключи, связывающие отдельные таблицы между собой информационными связями). Схема хранения данных, представляющих физическую модель БД, должна быть нормализована в соответствии с основными правилами реляционной алгебры.

В таблице 4 приведена спецификация таблиц базы данных АИС «Сибинтек: Управление заявками».

Таблица 4 – Спецификация БД АИС «Сибинтек: Управление заявками»

| Поле | Тип для СУБД MySQL | Назначение поля таблицы | Ключ |
|---|--------------------|---|------|
| Department – справочник подразделений Компании | | | |
| dep_id | Int | Идентификатор | PK |
| dep_name | Varchar(100) | Наименование подразделения | - |
| dep_phone | Varchar(20) | Телефон подразделения | - |
| Employee – справочник сотрудников, имеющих аккаунты в АИС «Сибинтек: Управление Заявками» | | | |
| emp_id | Int | Идентификатор | PK |
| emp_furname | Varchar(20) | Фамилия сотрудника | - |
| emp_firstname | Varchar(20) | Имя сотрудника | - |
| emp_secondname | Varchar(20) | Отчество сотрудника | - |
| emp_id_dep | Int | Идентификатор отдела | FK |
| emp_email | Varchar(20) | Адрес электронной почты | - |
| emp_post | Varchar(20) | Должность сотрудника | - |
| emp_login | Varchar(20) | Данные учетной записи: логин и пароль пользователя | - |
| emp_password | Varchar(20) | | - |
| emp_account | Int | Тип учетной записи пользователя: 0 – исполнитель, 1 – клиент-менеджер, 2 – руководитель, 3 – администратор | - |
| Status – справочник возможных статусов заявки | | | |
| sts_id | Int | Идентификатор | PK |
| sts_name | Varchar(15) | Наименование статуса | - |
| sts_color_r | Int | Компоненты цветовой модели (R,G,B), определяющие цвет заявки соответствующего статуса в системе | - |
| sts_color_g | Int | | - |
| sts_color_b | Int | | - |
| OrderType – справочник типов заявок | | | |
| ort_id | Int | Идентификатор | PK |
| ort_name | Varchar(100) | Наименование типа заявки | - |

Продолжение таблицы 4

| Поле | Тип для СУБД MySQL | Назначение поля таблицы | Ключ |
|---|--------------------|--|------|
| Service – справочник работ (средств), которые составляют наполнение заявок | | | |
| srv_id | Int | Идентификатор | PK |
| srv_name | Varchar(100) | Наименование работы | - |
| srv_price | Decimal | Стоимость | - |
| Client – справочник клиентов Компании | | | |
| cli_id | Int | Идентификатор | PK |
| cli_orgname | Varchar(50) | Наименование организации клиента | |
| cli_furname | Varchar(20) | Фамилия контактного лица | - |
| emp_firstname | Varchar(20) | Имя контактного лица | - |
| cli_secondname | Varchar(20) | Отчество контактного лица | - |
| cli_phone | Varchar(20) | Телефон контактного лица | - |
| cli_email | Varchar(20) | Адрес электронной почты контактного лица | - |
| Priority – справочник приоритетов заявок | | | |
| pri_id | Int | Идентификатор | PK |
| ri_name | Varchar(10) | Наименование приоритета | - |
| Поле | Тип для СУБД MySQL | Назначение поля таблицы | Ключ |
| pri_level | Int | Числовое выражение приоритета | - |
| pri_color_r | Int | Компоненты (R,G,B), определяющие цвет заявки соответствующего приоритета в системе | - |
| pri_color_g | Int | | - |
| pri_color_b | Int | | - |
| RouteRule – набор правил распределения заявок системой (устанавливает правила маршрутизации заявок) | | | |
| rtr_id | Int | Идентификатор | PK |
| rtr_name | Varchar(50) | Наименование правила | - |
| rtr_id_type | Int | Идентификатор типа заявки | FK |
| rtr_registration | Int | Тип регистрации заявки | - |
| rtr_id_client | Int | Идентификатор клиента | FK |
| rtr_id_priority | Int | Идентификатор приоритета заявки | FK |
| rtr_id_employee | Int | Идентификатор ответственного исполнителя | FK |
| TimeRule – набор правил распределения заявок системой (устанавливает правила определения нормативного срока решения заявок) | | | |
| dlr_id | Int | Идентификатор | PK |
| dlr_name | Varchar(50) | Наименование правила | - |
| dlr_id_type | Int | Идентификатор типа заявки | FK |
| dlr_registration | Int | Тип регистрации заявки | - |
| dlr_id_client | Int | Идентификатор клиента | FK |
| dlr_id_priority | Int | Идентификатор приоритета заявки | FK |
| dlr_normdays | Int | Нормативный срок решения заявки | - |
| Order – реестр созданных в системе заявок | | | |
| ord_id | Int | Идентификатор | PK |
| ord_registration | Int | Тип регистрации заявки (0 – по звонку менеджеру, 1 – онлайн регистрация) | - |
| ord_create | DateTime | Дата создания заявки | - |
| ord_description | Text | Описание заявки | - |
| ord_id_client | Int | Идентификатор клиента | FK |
| ord_id_priority | Int | Идентификатор приоритета заявки | FK |
| ord_deadline | Date | Дата срока решения заявки | - |
| ord_id_status | Int | Идентификатор статуса заявки | FK |

Продолжение таблицы 4

| Поле | Тип для СУБД MySQL | Назначение поля таблицы | Ключ |
|---|--------------------|--|--------|
| ord_id_executor | Int | Идентификатор ответственного исполнителя заявки | FK |
| ord_complete | Date | Дата решения заявки | - |
| ord_id_type | Int | Идентификатор типа заявки | FK |
| ApplyService – таблица учета выполненных работ по заявкам | | | |
| aps_id_service | Int | Идентификатор выполненной работы | PK, FK |
| aps_id_order | Int | Идентификатор заявки, для которой учитывается выполненная работа | FK, FK |
| aps_date | | Дата создания учетной записи по выполненной работе | - |
| aps_count | Int | Количество выполненных работ указанного типа | - |
| aps_note | Varchar(255) | Комментарий (пояснения) к записи о выполненной работе | - |
| Comment – комментарии к заявкам | | | |
| com_id | Int | Идентификатор | PK |
| com_id_order | Int | Идентификатор заявки, для которой оставляется комментарий | FK |
| com_date | DateTime | Дата оставления комментария | - |
| com_text | Text | Текст комментария | - |
| Поле | Тип для СУБД MySQL | Назначение поля таблицы | Ключ |
| com_image | Mediumblob | Двоичные данные изображения, прикрепленного к комментарию | - |
| com_id_employee | Int | Идентификатор сотрудника, оставившего комментарий к заявке | FK |

Анализ состава и структуры спроектированных таблиц БД АИС «Сибинтек: Управление заявками» позволяет заключить, что все таблицы находятся, как минимум, в третьей нормальной форме и не требуют дальнейшей нормализации.

Схема физической модели базы данных АИС «Сибинтек: Управление заявками», выполненная в соответствии с описанными таблицами, приведена на рисунке 13.

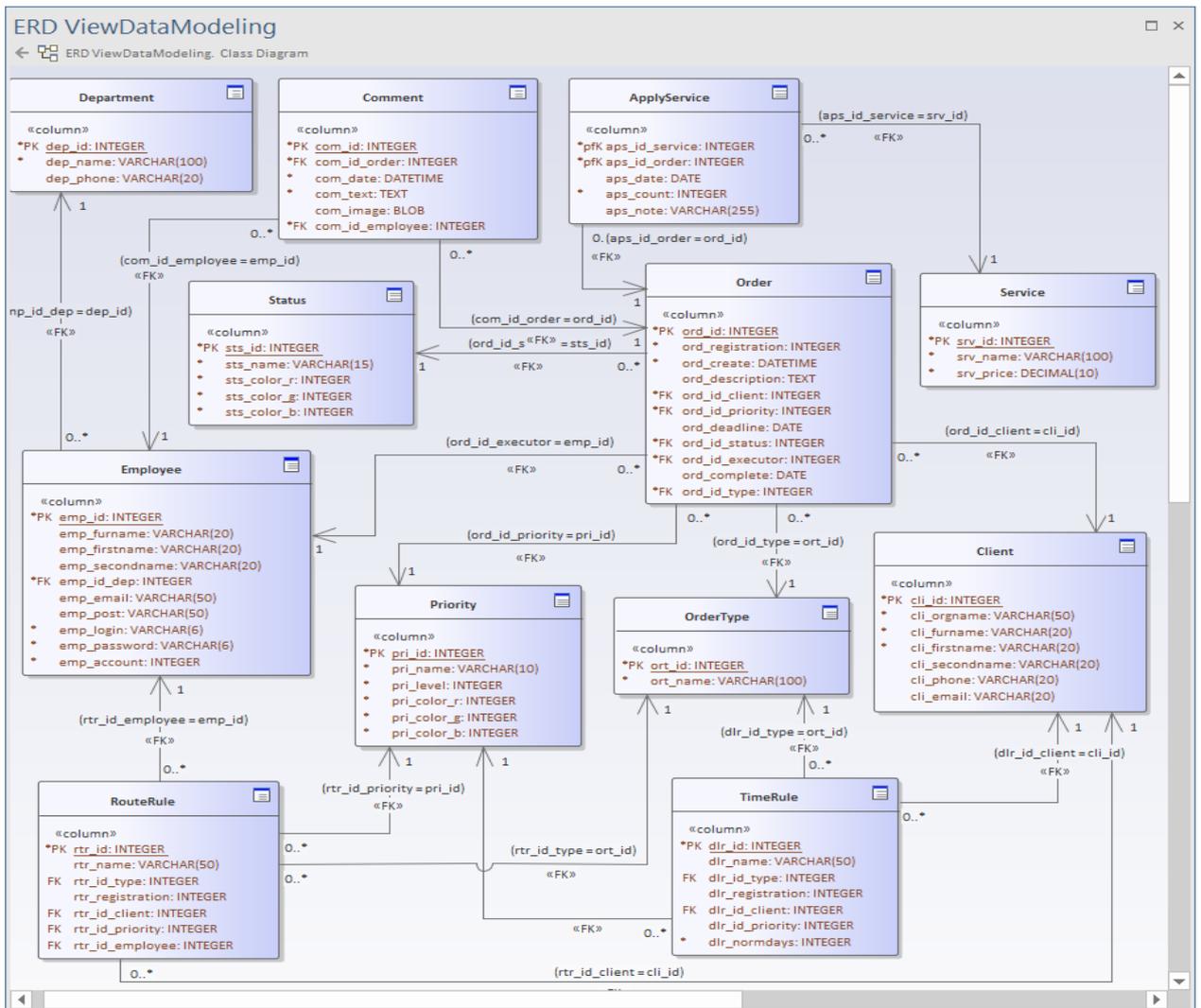


Рисунок 13 – Физическая модель базы данных АИС

Указанный на модели и в спецификации состав таблиц в базе данных системы является достаточным для реализации всех заявленных функций и решения комплекса поставленных задач. Средствами СУБД в базе данных реализована поддержка целостности данных за счет каскадного обновления при выполнении операций обновления или удаления записей.

Информация в базе данных защищена от несанкционированного доступа:

- процедурой авторизации в ИС для получения доступа к данным;
- паролем администратора базы данных на сервере СУБД;
- паролем администратора сервера СУБД.

2.5 Требования к аппаратно-программному обеспечению АИС «Сибинтек: Управление заявками»

В комплекс технических средств (КТС) АИС «Сибинтек: Управление заявками» входят следующие технические компоненты:

- компьютер менеджера по работе с клиентами с установленным исполняемым файлом АИС «Сибинтек: Управление заявками» и подключением к локальной сети (LAN) компании – рекомендуемая операционная система (ОС) – Microsoft Windows 8/8.1/10;

- компьютеры сотрудников-исполнителей с установленным исполняемым файлом АИС «Сибинтек: Управление заявками» и подключением к локальной сети (LAN) компании – рекомендуемая ОС – Microsoft Windows 8/8.1/10;

- компьютер администратора системы с установленным исполняемым файлом АИС «Сибинтек: Управление заявками» и подключением к локальной сети (LAN) компании – рекомендуемая ОС – Microsoft Windows 8/8.1/10;

- компьютеры руководителей исполнительных отделов с установленным исполняемым файлом АИС «Сибинтек: Управление заявками» и подключением к локальной сети (LAN) компании – рекомендуемая ОС – Microsoft Windows 8/8.1/10;

- сервер СУБД компании с размещенной на нем СУБД MySQL– рекомендуемая серверная ОС – Microsoft Windows Server 2016, в СУБД должна быть развернута база данных системы;

- устройства, обеспечивающие организацию сетевой инфраструктуры (локальной сети) компании: свитчи, хабы, линии связи – например, кабели типа витая пара категории 5е;

Таким образом, каждый пользователь будет работать с АИС на своем личном рабочем месте, но все данные и информация системы будет централизована на сервере БД.

Для сбора информации не применяются никаких специальных средств. Ввод информации в систему производится пользователями в процессе выполнения своих функций.

Для возможности работы (и доступа к БД) системные блоки рабочих мест пользователей АИС «Сибинтек: Управление заявками» должны быть укомплектованы сетевыми адаптерами, поддерживающими принятые настройки и спецификацию существующей локальной сети. IP-адреса сетевых адаптеров должны быть выбраны из пула адресов, принятых в компании на текущий момент времени.

Ручной ввод информации в АИС «Сибинтек: Управление заявками» осуществляется с помощью устройств ввода, таких, как: клавиатура манипулятор «мышь».

Вывод информации системой осуществляется на экран монитора рабочего компьютера пользователя или на принтер.

План организации и расположения рабочих мест пользователей АИС «Сибинтек: Управление заявками» должен быть выполнен в соответствии с действующими нормами и сводами правил, а также должен соответствовать правилам безопасности, эргономики и технической эстетики.

Работа в АИС «Сибинтек: Управление заявками» не требует применения новых технических средств. Все технические средства, требуемые для работы пользователей, уже имеются в наличии в компании. Рекомендуются обновить на компьютерах операционную систему Microsoft Windows до версии 10.

Выводы по главе 2: логическая модель АИС «Сибинтек: Управление заявками» показывает основные прецеденты системы и круг действующих лиц (пользователей): менеджер по работе с клиентами, сотрудники-исполнители, администратор, руководители исполнительных отделов.

Логическая модель позволила идентифицировать основные сущности рассматриваемой предметной области, что также дало возможность разработать информационную модель системы. Информационная модель

послужила основой для разработки схемы базы данных проектируемой АИС с подробной спецификацией.

Построенная модель данных АИС «Сибинтек: Управление заявками» соответствует требованиям нормальных форм и приведена, как минимум, к третьей нормальной форме.

Также были определены требования к комплексу технических средств АИС «Сибинтек: Управление заявками»: требования к рабочим станциям, их программному оснащению, сетевой инфраструктуре. Основываясь на этих требованиях, можно заключить, что существующая аппаратная база компании ООО «ИК «Сибинтек» удовлетворяет этим требованиям и позволяет в полном объеме развернуть и внедрить проектируемую АИС.

Глава 3 Физическое проектирование АИС «Сибинтек: Управление заявками»

3.1 Описание архитектуры АИС «Сибинтек: Управление заявками»

Сложность любой программной системы определяется ее архитектурой. По [7] архитектура системы представляет собой организационную структуру системы, в которой определены ее компоненты, их интерфейсы и концептуальные решения взаимодействия между ними.

Говоря о структуре сложной программной системы, принято говорить о компонентах этой системы как о составных частях ее архитектуры, таким образом подразумевая модульную архитектуру системы. При этом компонентом системы будет называться ее модульная часть, которая инкапсулирует ее содержимое, а само воплощение компонента является замещаемым в его окружении [14].

Одним из самых эффективных подходов к построению архитектуры системы является использование модульного принципа, то есть выделение простых структурных функциональных компонентов в отдельные модули: пакеты, классы и т.д. Таким образом создается иерархическая структура системы, которая облегчает процесс ее разработки на всех стадиях жизненного цикла.

Основой построения модульной архитектуры программной системы является ее декомпозиция. Этот процесс является универсальным решением для снижения сложности системы – разбить сложную систему на множество связанных частей.

Структурная схема пакетов прикладного приложения обрисовывает его архитектуру в рамках использования, состава и взаимодействия программных модулей. Основные компоненты (классы, компоненты и пакеты) архитектуры верхнего уровня программного проекта АИС

«Сибинтек: Управление заявками» приведены на рисунке 14 на UML-диаграмме пакетов.

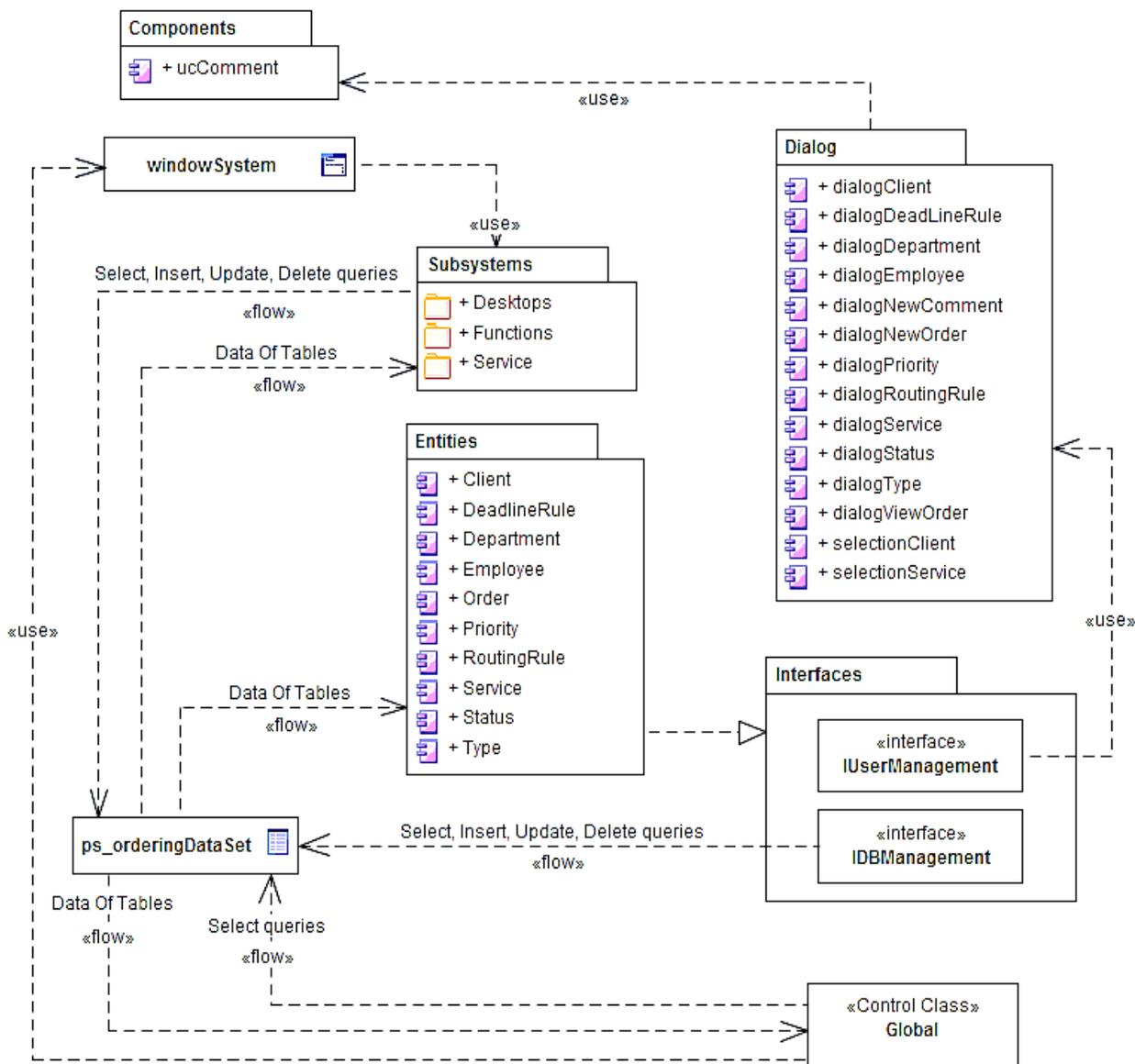


Рисунок 14 – Диаграмма пакетов АИС «Сибинтек: Управление заявками»

Основные компоненты (классы, компоненты и пакеты) архитектуры верхнего уровня программного проекта АИС «Сибинтек: Управление заявками» включают в себя:

- пакет **Components** содержит пользовательские компоненты интерфейса для локальных задач;

- пакет Subsystems – набор подсистем пользователя с реализацией всех основных функций АИС;
- пакет Dialog – содержит набор форм ввода данных, посредством которых организовывается диалог с пользователем;
- Entities – пакет, представляющий набор классов-сущностей, которые предоставляют возможности реализации функций, поддерживают управление данными и взаимодействие с БД;
- пакет Interfaces – набор интерфейсов для осуществления доступа сущностей к данным в базе данных и управления сущностями пользователями через диалоговые окна (пакет Dialog);
- windowSystem – класс главной формы системы;
- Global – головной класс системы, содержащий методы и поля, являющиеся глобальными для всей системы, включая авторизацию и данные о текущем пользователе, авторизованном в системе;
- ps_orderingDataSet – компонент организации доступа классов сущностей к базе данных, организованный через провайдеров подключения к наборам данных, отображаемых из таблиц физической базы данных.

3.2 Выбор технологии разработки программного обеспечения АИС «Сибинтек: Управление заявками»

Разработка проекта программного обеспечения АИС «Сибинтек: Управление заявками» выполняется в соответствии с методологией графического описания систем с разработкой комплекса моделей на унифицированном языке графического моделирования UML.

Для разработки моделей всех видов с использованием языка UML применяются специальные программные средства – CASE-системы. Современный рынок программного обеспечения насыщен множеством различных CASE-систем разных производителей. Из наиболее популярных CASE-систем можно выделить следующие: Rational Rose, Power Designer,

StarUML, Enterprise Architect и другие. Все эти продукты полностью поддерживают нотацию UML 2.0 и предлагают широкие возможности и инструменты по созданию моделей различных типов.

В таблице 5 приведена сравнительная характеристика наиболее распространенных CASE-систем.

Таблица 5 – Сравнительная характеристика CASE-систем

| Возможность | Rational Rose | Power Designer | StarUML | Enterprise Architect | Visual Paradigm for UML |
|------------------------------|----------------------|-----------------------------------|--------------------|---|--|
| Прототипирование интерфейса | Нет | Нет | Нет | Да | Да |
| Forward Engineering (для C#) | Нет | Да | Да | Да | Да |
| Reverse Engineering (для C#) | Нет | Да | Да | Да | Да |
| Модель требований | Нет | Да | Нет | Да | Да |
| Нотации моделирования | UML, IDEF1X | BPMN, EPC, UML, IDEF1X, ODM, NIEM | UML, IDEF1X, DFD | BPMN, EPC, UML, IDEF1X, DFD, ODM, SPEM, GML | BPMN, EPC, UML, IDEF1X, EPC, EGB, WDSL |
| Лицензия | Платная | Платная | Условно бесплатная | Платная | Платная |

Из приведенных в таблице 5 CASE-систем, для проектирования программного обеспечения АИС «Сибинтек: Управление заявками» была выбрана Enterprise Architect (версия 15.0). По мнению автора этот продукт наиболее удобен в использовании. Также в данной системе созданы и все остальные модели системы (организационная, требования, функциональная структура и т.д.).

В дополнение Enterprise Architect поддерживает функции Database Forward Engineering, позволяющие автоматически генерировать скрипт создания физической структуры базы данных по построенной модели данных с учетом выбранной СУБД.

Для быстрой и эффективной разработки приложений применяются специальные среды разработки – IDE (Integrated Development Environment), которые включают оптимизированные наборы инструментов: редакторы визуальных форм, палитры компонентов для построения пользовательского интерфейса, редактор программного кода [5].

Для разработки и проектирования программного обеспечения АИС «Сибинтек: Управление заявками» выбрана среда разработки Microsoft Visual Studio 2015. Данная система предлагает разработчику:

- редактор форм;
- редактор кода;
- средства инжиниринга и разработки баз данных;
- мощные инструменты для анализа структуры кода;
- мощные средства рефакторинга исходного кода.

Таким образом, выбранная среда разработки имеет набор компонентов, ускоряющих разработку и облегчающих интеграцию в приложение баз данных, что в конечном итоге позитивно влияет на эффективность и скорость разработки, уменьшает вероятность ошибок на этапе сборки приложения.

Для написания исходного кода приложения АИС «Сибинтек: Управление заявками» выбран язык программирования высокого уровня C#.NET. Этот язык программирования в настоящее время является одним из фундаментальных для разработки бизнес-приложений (наряду с Java, Python, C++). Средства C# главным образом способствуют повышению продуктивности разработчиков с помощью реализованного в языке баланса между простотой, выразительностью и производительностью [13]. Язык C# предлагает такие преимущества разработчику, как:

- унификация системы типов – наследование всех объектов от базового типа – Object;
- поддержка парадигмы ООП и применения интерфейсов, замещающих устаревшие схемы с использованием множественного

наследования (которые уже не поддерживают многие ведущие языки программирования);

- поддержка делегатов, позволяющих использовать объекты типа функций и делегировать управление;
- использование декларативных шаблонов [13];
- поддержка лямбда-выражений;
- поддержка Entity framework и встроенных Linq-запросов.

Таким образом, для разработки ЭОТС выбран стек программных инструментов, включающих C# .NET Visual Studio 2015 и CASE Enterprise Architect 15.0.

3.3 Выбор СУБД АИС «Сибинтек: Управление заявками»

Для хранения данных информационной системы требуется организация базы данных, что подразумевает применение СУБД.

СУБД – это программные средства, с помощью которых можно создавать базы данных, наполнять их и работать с ними [25]. Многие существующие СУБД являются специализированными языками программирования. Для СУБД определены следующие функциональные требования [10]:

- СУБД должна допускать определения данных и иметь интерпретатор языка определения данных (ЯОД);
- СУБД должна обрабатывать пользовательские запросы на селекцию или манипуляции данными и быть оснащена интерпретатором языка манипулирования данными (ЯМД);
- СУБД должна обеспечивать поддержку целостности данных, контролируя и блокируя попытки нарушить установленные ограничения;
- СУБД должна обеспечивать восстановление после сбоев;

- СУБД должна поддерживать словарь данных – наличие системной базы данных, которая содержит сведения о всех пользовательских БД, созданных в этой СУБД, хранит важные переменные среды;

- СУБД должна иметь высокую производительность при выполнении своих функций, рассчитанной на ее одновременное использование многими клиентами.

Всем этим требованиям соответствует СУБД MySQL, которая является наиболее распространенной свободно-распространяемой системой управления базами данных [19]. Данная СУБД обладает высокой эффективностью и высокой надёжностью, что способствовало столь высокой популярности данной СУБД. Основные достоинства СУБД MySQL:

- многопоточность, поддержка нескольких одновременных запросов;
- оптимизация связей;
- записи фиксированной и переменной длины;
- наличие ODBC драйвера, в том числе и для MS Visual Studio;
- гибкая поддержка форматов, строк переменной длины и меток времени;
- быстрая работа, масштабируемость;
- совместимость с ANSI SQL;
- распространяется согласно стратегии Open Source;
- хорошая поддержка со стороны провайдеров услуг хостинга.

Еще одним немаловажным достоинством СУБД MySQL является ее интеграция с системой разработки и программирования MS Visual Studio. Средства MS Visual Studio и языка программирования высокого уровня C# предлагают широкий набор средств для интеграции баз данных непосредственно в систему программирования, что существенно ускоряет процесс разработки и делает систему надежнее [2].

3.4 Реализация физической модели базы данных АИС «Сибинтек: Управление заявками»

Схема физической модели базы данных АИС «Сибинтек: Управление заявками», спроектированная в п. 2.4, соответствует требованиям нормальных форм. На практике не всегда целесообразно максимально нормализовать переменные отношения. Как правило, для хорошей модели данных достаточно довести схему отношений до третьей нормальной формы включительно [21]. Анализ состава и структуры спроектированных таблиц БД АИС «Сибинтек: Управление заявками» позволяет заключить, что все таблицы находятся, как минимум, в 3НФ и не требуют дальнейшей нормализации. Таким образом, полученная модель может быть реализована в виде физических таблиц в выбранной СУБД.

Приведенный состав таблиц базы данных (таблица 4) позволяет хранить всю необходимую справочную и оперативную информацию для функционирования АИС «Сибинтек: Управление заявками» и решения комплекса поставленных задач.

Реализация физической модели может быть автоматизирована средствами Enterprise Architect. Для этого в среде этой CASE-системы создается отдельное подключение к предварительно созданному файлу базы данных.

Далее в CASE-системе выбирается список объектов, которые будут сгенерированы в новом файле базы данных (как показано на рисунке 15).

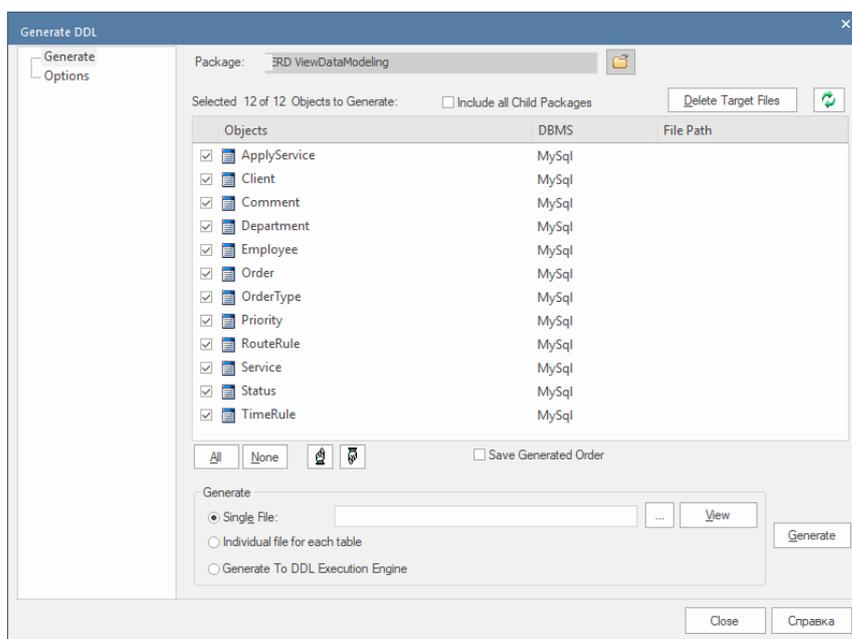


Рисунок 15 – Настройка генерации в системе Enterprise Architect

При этом после выполнения команды «Generate» система сгенерирует исходный код формирования физической структуры базы данных, записанный на языке описания данных DDL (Data Description Language). Этот код впоследствии выполняется в среде выбранной СУБД как простой запрос.

В приложении А приведен сгенерированный в Enterprise Architect DDL-код базы данных АИС «Сибинтек: Управление заявками».

Любая база данных должна обеспечивать полноту и достоверность данных. Ценность этой информации образуется в процесс эксплуатации БД с учетом осмысленного оперирования данными пользователями, которые не способны работать безошибочно ввиду влияния человеческого фактора.

Поскольку целостность данных может нарушаться только операциями обновления (удаления), достаточно при обновлении (удалении) данных обеспечить соблюдение правил делового регламента [24].

Поддержку целостности данных можно автоматизировать средствами примененной СУБД. Для этого деловой регламент необходимо представить формально в виде совокупности предикатов, определённых на объектах БД.

При любой попытке обновления состояния базы данных должна автоматически запускаться процедура вычисления значений ограничений целостности данных. Обновлённое состояние БД фиксируется в только после того, как СУБД убедится в его целостности, т.е. в истинности всех значений предикатов.

Такая идея реализуется практически во всех современных СУБД, в том числе и в MySQL. При этом каждое ограничение целостности данных может настраиваться в соответствующих отношениях модели данных – как на этапе проектирования БД (с применением средств автоматизированного проектирования), так и на этапе ее реализации и настройки в СУБД.

На рисунке 16 приведено диалоговое окно настройки ограничения целостности данных на примере связи, определяющей аккаунт собственника. В данном случае данное ограничение установлено на каскадное удаление (обновление) записей в подчиненной таблице при их модификации в родительской.

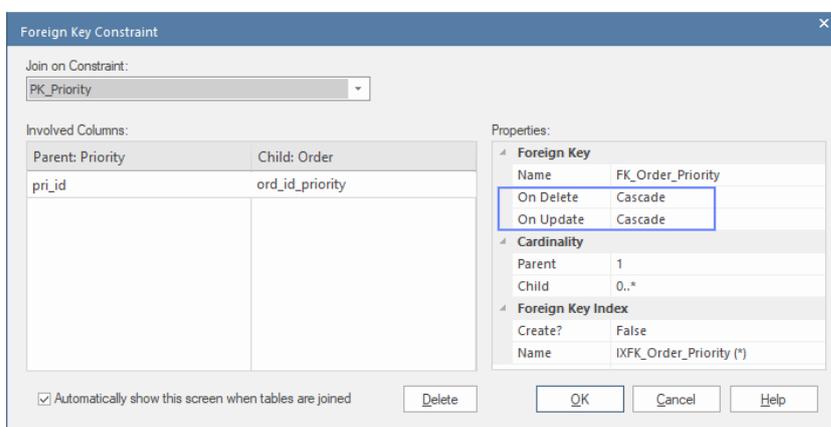


Рисунок 16 – Настройка внешнего ключа в системе Enterprise Architect

На рисунке 17 приведен фрагмент реализации этого ограничения в СУБД. Как видно, все настройки по ограничению целостности данных, выполненные на этапе проектирования в Enterprise Architect, успешно экспортировались в физическую реализацию базы данных в СУБД.

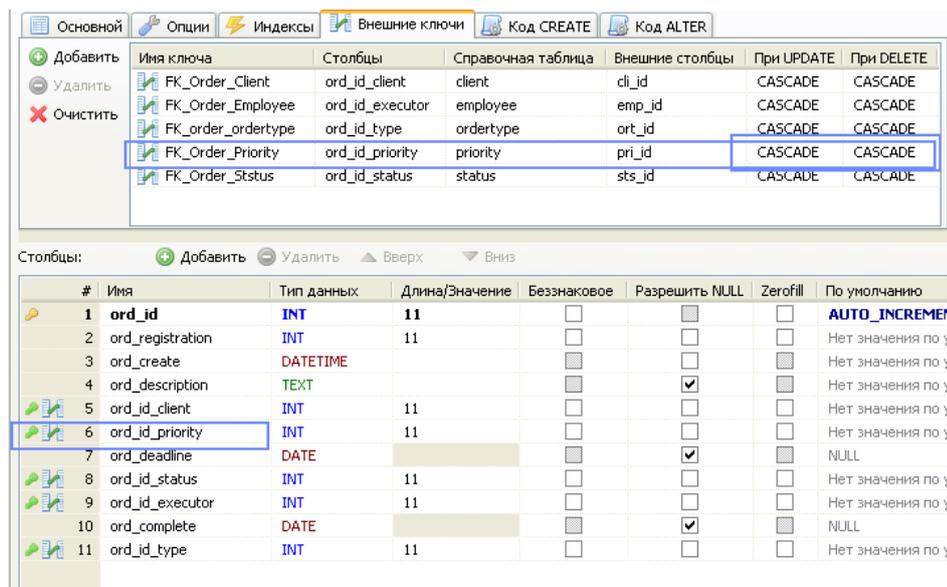


Рисунок 17 – Свойства внешнего ключа в среде MySQL (Heidi SQL)

Как видно из рисунка 17, ограничения, установленные в CASE-системе, корректно экспортировались в физическую модель базы данных.

Таким образом, поддержка целостности данных таблиц в БД АИС «Сибинтек: Управление заявками» обеспечивается средствами СУБД за счет каскадного обновления при выполнении операций обновления или удаления записей.

Информация в БД защищена от несанкционированного доступа процедурой авторизации для получения доступа к данным и паролем суперпользователя (администратора) БД на сервере СУБД.

3.5 Разработка программного обеспечения АИС «Сибинтек: Управление заявками»

Функциональная структура АИС «Сибинтек: Управление заявками» включает в себя основные и служебные функции. Набор функций АИС «Сибинтек: Управление заявками» приведен на рисунке 18.



Рисунок 18 – Дерево функций АИС «Сибинтек: Управление заявками»

В составе основных функций АИС «Сибинтек: Управление заявками» присутствуют функции, которые выполняют непосредственно автоматизацию обработки заявок, а также здесь собраны вспомогательные решения и подсистемы, косвенно влияющие на выполнение функций и (или) предоставляющие для этого средства и данные:

- ведение вспомогательных справочников;
- регистрация и учет заявок клиентов с возможностью установки приоритета;
- учет выполнения заявок клиентов посредством установки статуса заявок (ведение вспомогательного справочника значащих статусов);
- учет выполняемых работ по заявкам клиентов (добавление выполненных работ, примененных средств, комментирование работ и этапов хода выполнения заявки, расчет стоимости заявки);
- автоматизированная обработка заявок (разработка и применение правил маршрутизации заявок, по которым заявка назначается ответственному исполнителю; разработка и применение правил установки сроков решения заявок – установка дедлайна для ответственных исполнителей и контроль сроков решения заявки);

- составление сводных отчетов для руководителя (статистика, финансовые данные, динамика решения заявок).

В число служебных функций системы входят функции, которые так или иначе обслуживают выполнение основных функций. К служебным функциям АИС «Сибинтек: Управление заявками» относятся:

- авторизация пользователя в;
- организация доступа к данным и функциям системы, определяющая права доступа авторизованного пользователя к набору функций системы и возможности управления данными;
- проверка подключения к базе данных;
- установление связи с таблицами БД посредством специальных адаптеров;
- управление данными, в результате которого информация, введенная пользователем, отражается в физических таблицах БД;
- контроль вводимых данных, обеспечивающий проверку информации, вводимой пользователем;
- обработка исключительных ситуаций, возникающих в программе.

Диалог системы с пользователями организован через главное меню – для каждого типа пользователя определен свой состав главного меню (рисунок 19).

Меню администратора составляет перечень подсистем управления вспомогательными справочниками.

Меню клиент-менеджера предоставляет доступ к подсистеме учета заявок, справочнику клиентов, справочнику сотрудников (только просмотр), подсистеме разработки правил автоматизированной обработки заявок.

Меню исполнителя содержит учетную таблицу «своих» заявок и возможности учета их выполнения.

Меню руководителя предоставляет доступ к подсистемам формирования сводных отчетов. Руководитель также может создавать правила автоматизированной обработки заявок и просматривать все заявки.

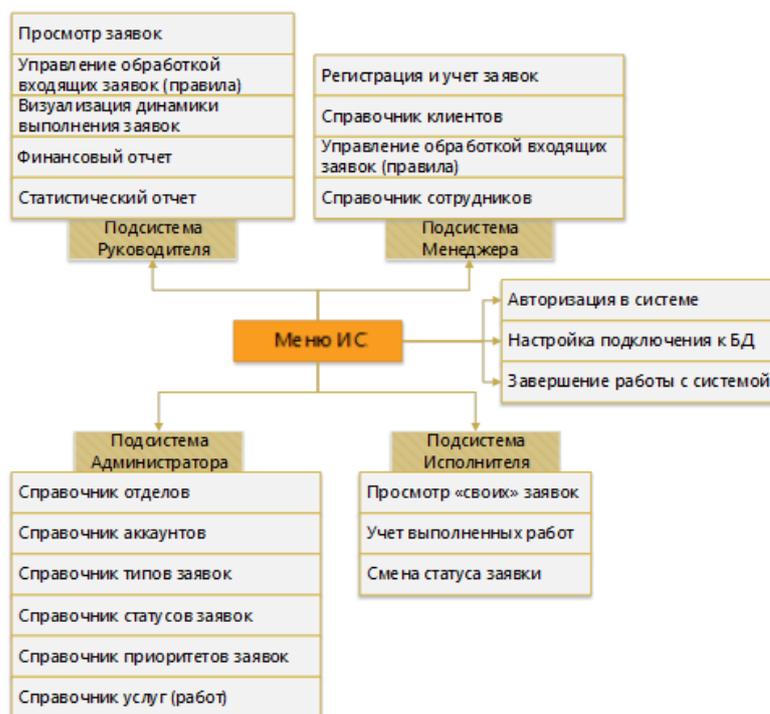


Рисунок 19 – Сценарий диалога АИС «Сибинтек: Управление заявками»

Все функции системы, как было установлено в п.3.1, вынесены в отдельные программные модули. Имеет смысл описать каждый модуль системы.

Основным пакетом в АИС «Сибинтек: Управление заявками» является пакет Entities – он является «ядром» всей системы в части управления данными. Диаграмма компонентов пакета Entities приведена на рисунке 20.

Классы-сущности реализуются для сущностей (объектов) предметной области, данные о которых хранятся в БД. В АИС «Сибинтек: Управление заявками» такими сущностями являются клиенты (класс Client), заявки (класс Order), статусы заявок (класс Status), услуги (класс Service), подразделения Компании (класс Department), сотрудники Компании, имеющие аккаунты в АИС «Сибинтек: Управление заявками» (класс Employee), типы входящих заявок (класс Type), приоритеты входящих заявок (класс Priority), правила автоматизированной обработки заявок – по определению дедлайна (класс DeadlineRule) и при маршрутизации заявки (класс RoutingRule).

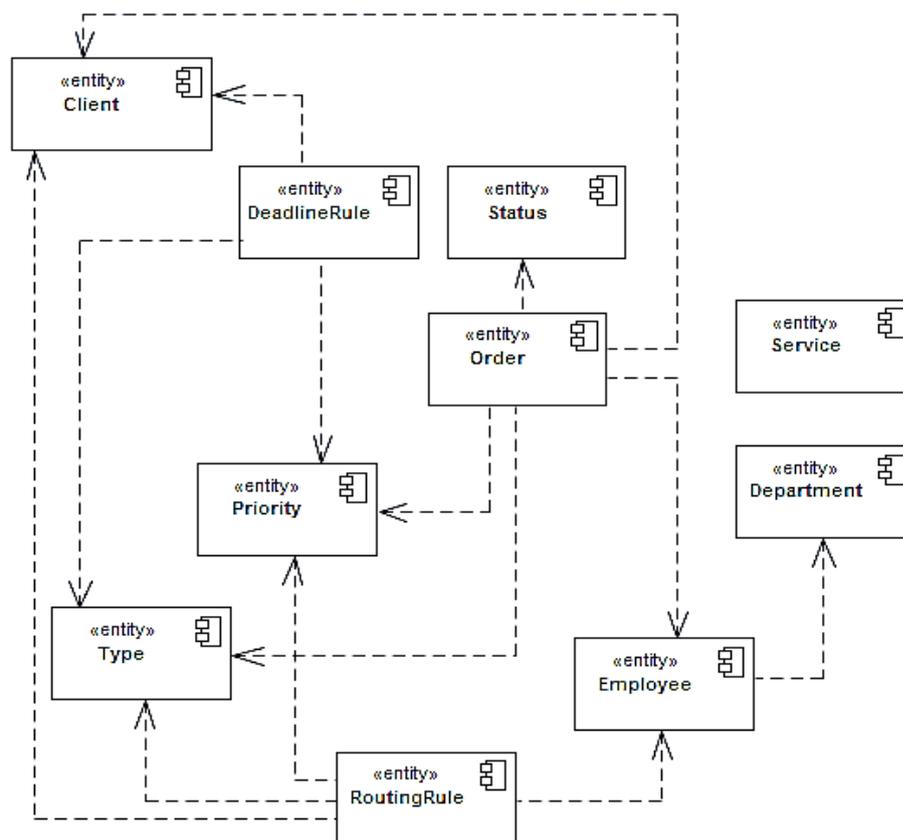


Рисунок 20 – Диаграмма компонентов пакета Entities

Пакет Dialog обеспечивает систему компонентами (формами), посредством которых пользователь может управлять данными (справочников, оперативных таблиц). Формы данного пакета являются модальными диалогами ввода (редактирования) данных, поддерживающими также проверку корректности ввода данных пользователем.

Диалоги ввода/редактирования данных должны быть представлены для каждого класса-сущности. Для сущности «заявка» (Order) представлено два диалога – dialogNewOrder (позволяет вводить первичные данные входящей заявки в систему) и dialogViewOrder (позволяет осуществлять сопровождение и учет решения заявки).

Помимо диалогов для классов-сущностей, в системе также представлены диалоги выбора записи из справочников (selectionService – выбор услуги (работы), selectionClient – выбор клиента) и подстановки

выбранного значения в соответствующие поля. Диаграмма компонентов пакета Dialog приведена на рисунке 21.

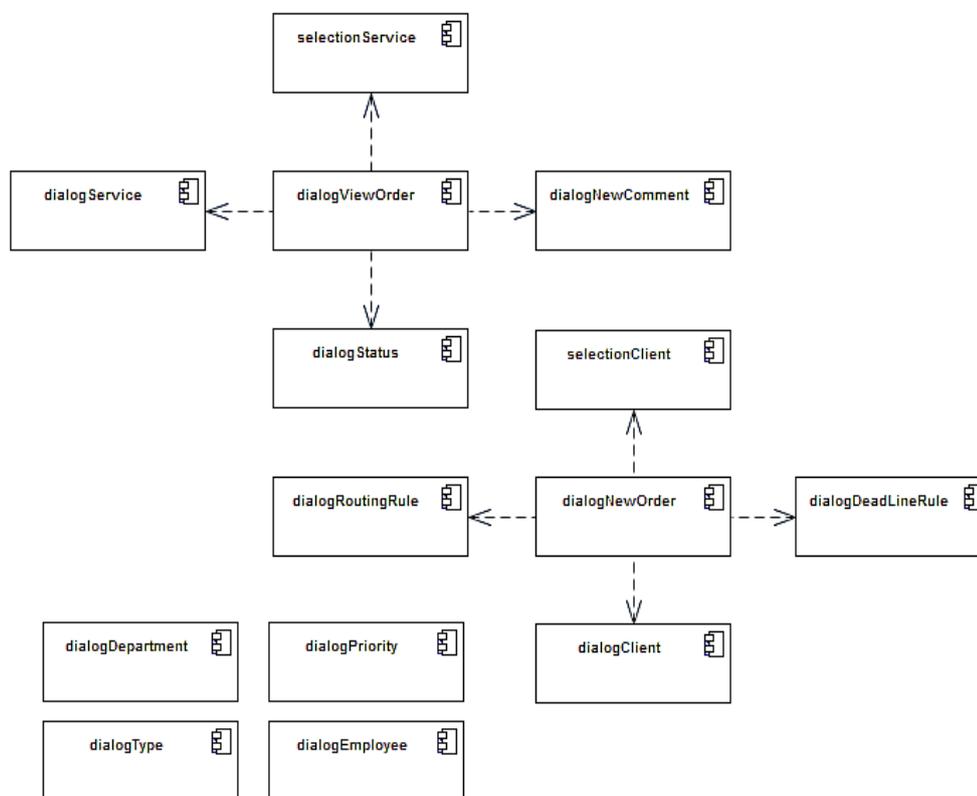


Рисунок 21 – Диаграмма компонентов пакета Dialog

Пакет Subsystems содержит в себе пакеты, составляющие основные функциональные возможности системы. Среди них пакеты:

– Desktops – пакет реализации рабочих столов пользователей системы;

– Functions – пакет основных функциональных модулей системы;

– Service - пакет вспомогательных (сервисных) модулей системы.

Диаграмма пакета Subsystems приведена на рисунке 22.

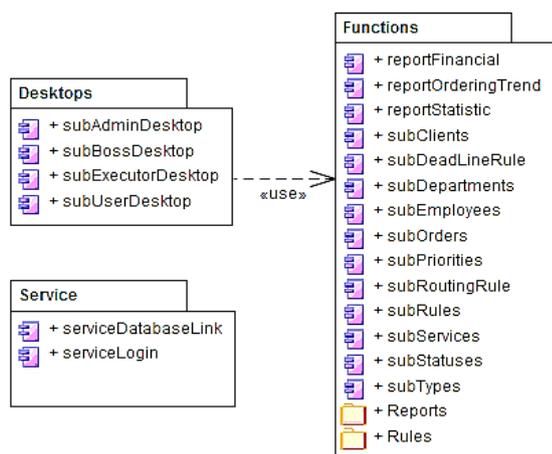


Рисунок 22 – Диаграмма компонентов пакета Subsystems

Пакет Functions представляет компоненты, позволяющие реализовывать основные функции системы, и содержит:

- subClients – модуль подсистемы ведения справочника клиентов;
- subEmployees – модуль подсистемы ведения справочника сотрудников Компании, имеющих в системе аккаунты;
- subDepartments – модуль подсистемы ведения справочника подразделений Компании;
- subOrders – модуль подсистемы ведения и учета реестра заказов и их выполнения (выполненных работ, комментирования и т.д.);
- subPriorities – модуль подсистемы ведения справочника приоритетов входящих заявок;
- subRules – модуль подсистемы ведения справочника правил автоматизированной обработки заявок (выделен в состав отдельного подпакета Rules: по маршрутизации – subRoutingRule и по расчету нормативного срока решения заявки – subDeadLineRule);
- subServices – модуль подсистемы ведения справочника услуг (работ) или средств, примененных и выполненных в процессе решения заявки;

– subStatuses – модуль подсистемы ведения справочника статусов заявок;

– subTypes – модуль подсистемы ведения справочника типов входящих заявок;

Диаграмма компонентов пакета Functions в соответствии с вышеописанным составом пакета приведена на рисунке 23.

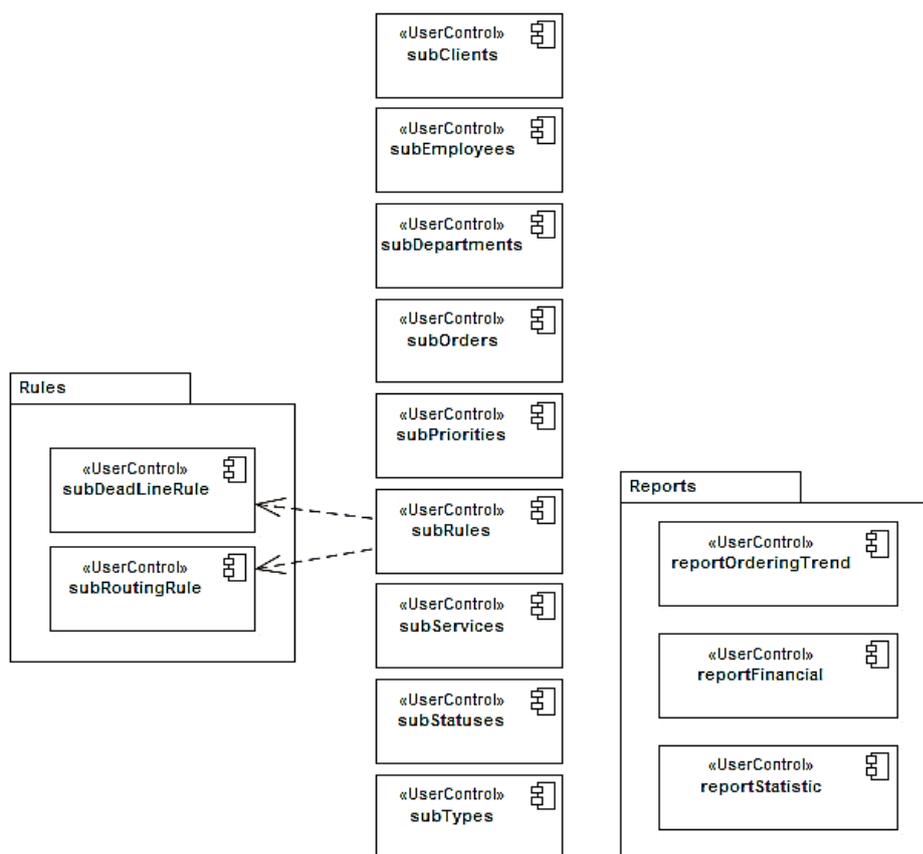


Рисунок 23 – Диаграмма компонентов пакета Functions

Пакет Desktops содержит модули рабочих столов для каждого типа пользователей, включая индивидуальное меню пользователя, скомпонованное в соответствии с заданными привилегиями и полномочиями пользователя.

Компоненты (рисунок 24), реализующие рабочие столы пользователей:

- subAdminDesktop – рабочий стол администратора АИС «Сибинтек: Управление заявками»;
 - subExecutorDesktop – рабочий стол администратора АИС «Сибинтек: Управление заявками»;
 - subBossDesktop – рабочий стол администратора АИС «Сибинтек: Управление заявками»;
 - subUserDesktop – рабочий стол администратора АИС «Сибинтек: Управление заявками».
-

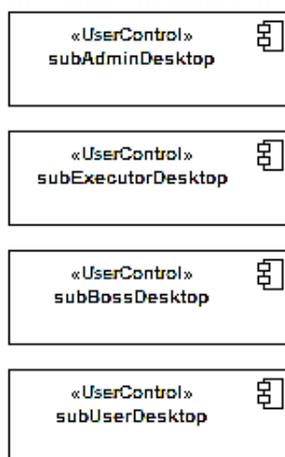


Рисунок 24 – Диаграмма компонентов пакета Desktops

Пакет Service представляет модули выполнения сервисных функций системы и содержит следующие компоненты (рисунок 25):

- панель авторизации пользователя в системе с наделением авторизованного пользователя правами в соответствии с уровнем доступа его аккаунта;
- панель настройки подключения системы к БД и проверки связи с ней.

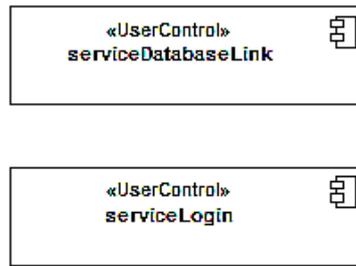


Рисунок 25 – Диаграмма компонентов пакета Service

3.6 Тестирование программного проекта

На этапе тестирования проверяется правильность работы программы, а также ее устойчивость – величина, характеризующая способность программного продукта сохранять нормальную работоспособность при отклонении условий функционирования различной степени (например, ввод некорректных данных) – в этих случаях в программу намеренно вводится противоречивая информация и анализируется ее реакция. Программа считается устойчивой, если она прошла все соответствующие тесты успешно. В дополнение к этим, также проводятся тесты, выявляющие удобство использования программы. Все ошибки и доработки, выявленные на данном этапе, устраняются – так выполняется отладка программы.

Полный перечень форм разработанной системы в процессе тестового прогона приведен в приложении Б.

Все подсистемы испытываются одновременно на корректность взаимодействия подсистем, влияние подсистем друг на друга, т.е. испытания проводятся комплексно.

Тестирование включают проверку:

- полноты и качества реализации функций;
- выполнения требований, относящихся к интерфейсу системы;
- работы пользователей в диалоговом режиме;

- полноты действий, доступных пользователю, и их достаточность для функционирования системы;
- сложности процедур диалога, возможности работы пользователей без специальной подготовки;
- реакции системы на ошибки пользователя.

В рамках тестирования разработанной АИС будут протестированы ее функции и выполнена оценка фактических результатов.

Первым этапом следует выполнить в системе создание правил маршрутизации заявок при автоматизированной обработке – в данном примере создается правило следующего типа: «Заявки по установке СКУД должны направляться на исполнение главному специалисту по СКУД Панюшкину В.А.» (рисунок 26).

Рисунок 26 – Создание нового правила маршрутизации заявок

На рисунке 27 приведен результат выполнения данного действия: в справочнике учета правил маршрутизации заявок появилось новое правило.

| Заявки | | | | | | |
|--|----------------------------------|--------------------|--------|-----------|----------------|--|
| + Создать... ✎ Редактировать... ✖ Удалить | | | | | | |
| Имя правила | Тип заявки | Способ регистрации | Клиент | Приоритет | Ответственный | |
| Установка СКУД | Установка СКУД | Любой | Любой | Любой | Панюшкин В. А. | |
| Интеграция систем анализа данных | Интеграция систем анализа данных | Любой | Любой | Любой | Зорин Е. В. | |
| | Разработка и | | | | | |

Рисунок 27 – Новое правило маршрутизации заявок в справочнике

Следующим этапом следует выполнить в системе создание правила срока исполнения заявок при автоматизированной обработке – в данном примере создается правило следующего типа: «Срочные заявки по установке СКУД должны быть обработаны и решены в течение 10 дней после их создания в системе» (рисунок 28).

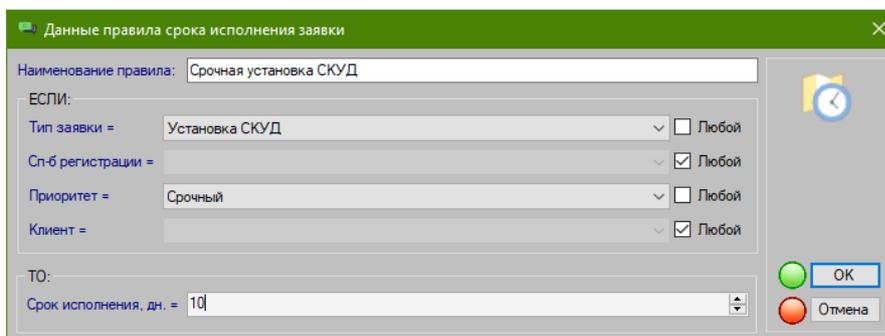


Рисунок 28 – Создание нового правила срока исполнения заявок

На рисунке 29 приведен результат выполнения данного действия: в справочнике учета правил срока исполнения заявок появилось новое правило.

| | | | | | | |
|---|----------------|-------|-------|---------|-------|-----|
| приоритетом | любой | любой | любой | любой | любой | 100 |
| Отложенные заявки (вспомнить через год) | Любой | Любой | Любой | Отложен | | 365 |
| Срочная установка СКУД | Установка СКУД | Любой | Любой | Срочный | | 10 |

Рисунок 29 – Новое правило срока исполнения заявок в справочнике

Теперь можно приступить к созданию новой заявки, которая будет соответствовать новым созданным правилам (рисунок 30).

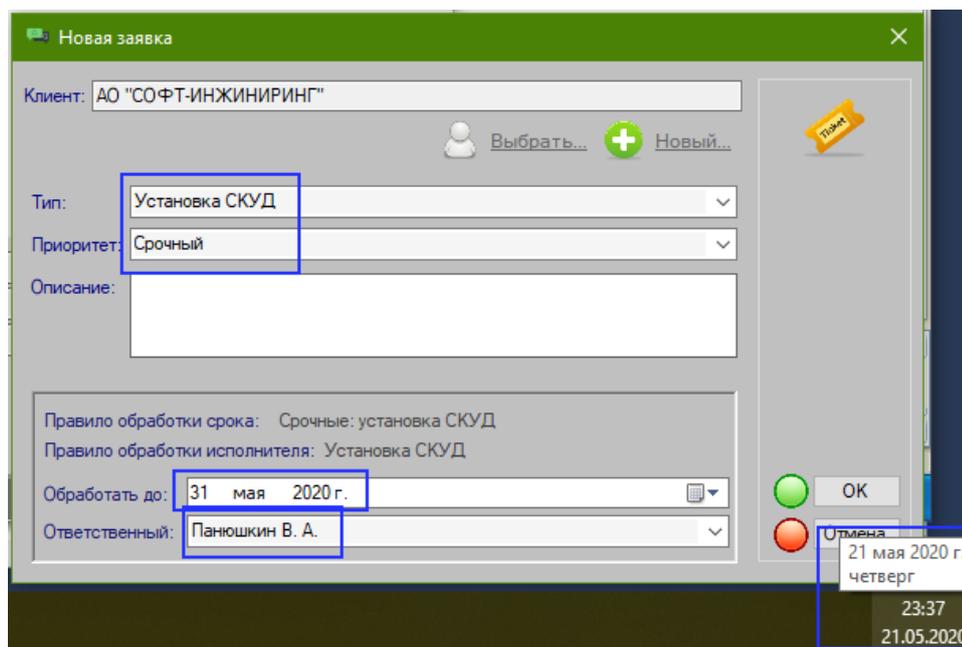


Рисунок 30 – Создание новой заявки

Из рисунка видно, что при выборе типа и приоритета заявки автоматически были идентифицированы подходящие правила обработки заявок – был установлен срок решения (заявка создана 21 мая, а срок решения был установлен 31 мая, то есть ровно 10 дней, как в созданном правиле) и ответственный исполнитель.

На рисунке 31 приведен результат выполнения действия по созданию тестовой заявки: в системе учета заявок срока появилась новая заявка с идентификационным номером 0020.

| | ID | Тип | Создана | Закрыта | Клиент | Статус | Решить | Тип регистр. | Ответств. |
|------------|------|--|------------|---------|--------------------------|-----------|------------|-----------------|-------------------|
| Заявки | 0002 | Установка СКУД | 23.05.2020 | | ИП Максим | Новая | 23.06.2020 | Служба менеджер | Панюшкин В. А. |
| Слиенты | 0020 | Установка СКУД | 21.05.2020 | | АО "СОФТ-ИНЖИНИРИНГ" | Новая | 31.05.2020 | Служба менеджер | Панюшкин В. А. |
| Правила | 0019 | Разработка и внедрение информационных систем | 28.05.2020 | | ООО "Феникс Контакт Рус" | В работе | 08.06.2020 | Служба менеджер | Белобородов К. В. |
| сотрудники | 0007 | Разработка и внедрение информационных систем | 24.05.2020 | | ООО "Феникс Контакт Рус" | Завершена | 28.06.2020 | Служба менеджер | Белобородов К. В. |

Рисунок 31 – Новая заявка в системе учета

Из рисунка 32 также видно, что новая заявка появилась в подсистеме у выбранного ответственного исполнителя.

| ID | Тип | Создана | Закрыта | Клиент | Статус | Решить | Тип регистрац | Ответственн |
|------|----------------|------------|---------|----------------------|--------|------------|-------------------|----------------|
| 0002 | Установка СКУД | 23.05.2020 | | ИП Максим | Новая | 23.06.2020 | Служба менеджеров | Панюшкин В. А. |
| 0020 | Установка СКУД | 21.05.2020 | | АО "СОФТ-ИНЖИНИРИНГ" | Новая | 31.05.2020 | Служба менеджеров | Панюшкин В. А. |

Рисунок 32 – Новая заявка в системе учета в подсистеме исполнителя

Следующим шагом тестирования заявки будет ее наполнение: добавление произведенных работ, создание комментариев (рисунок 33).

Заявка №0020
 Создана: 21 мая 2020 г.
 Обработать до: 31 мая 2020 г.
 Закрыта: -
 Клиент: АО "СОФТ-ИНЖИНИРИНГ"
 Попова А. В., т.: 89110002200; mail: alporova@engineering
 Тип заявки: Установка СКУД
 Приоритет заявки: Срочный
 Ответственный: Панюшкин В. А., т.: 795564
 Специалисты по СКУД: Главспец по СКУД

Комментарии

Панюшкин В. А. 21.05.2020 23:45
 Согласовали с заказчиком производителем и интерфейсом 2 уровня со СКУД

Выполненные работы / учтенные средства

| Дата | Работа / примененное оборудование | Стоим. за единицу | Кол-во | Сумма |
|------------|------------------------------------|-------------------|--------|----------|
| 21.05.2020 | Монтаж и подключение блока питания | 750 Р | 2 | 1 500 Р |
| 21.05.2020 | Блок питания БП 12/2А | 712 Р | 2 | 1 424 Р |
| 21.05.2020 | Монтаж КТС информационных систем | 14 500 Р | 1 | 14 500 Р |
| 21.05.2020 | Инсталляция ПО СКУД Равес | 4 000 Р | 1 | 4 000 Р |

Записей загружено: 4 Всего на сумму: 21 424 Р

Сменить статус: Новая

Рисунок 33 – Учет выполнения заявки

По мере того, как в историю выполнения заявки добавляются выполненные работы, автоматически обновляется ее полная расчетная стоимость.

Еще одним важным пунктом тестирования системы является проверка составления отчетной и сводной информации о выполнении заявок. Данная

информация доступна руководителю, поэтому при тестировании следует авторизоваться под учетной записью руководителя.

На рисунке 34 приведен результат построения сводного тренда по динамике обработки заявок.

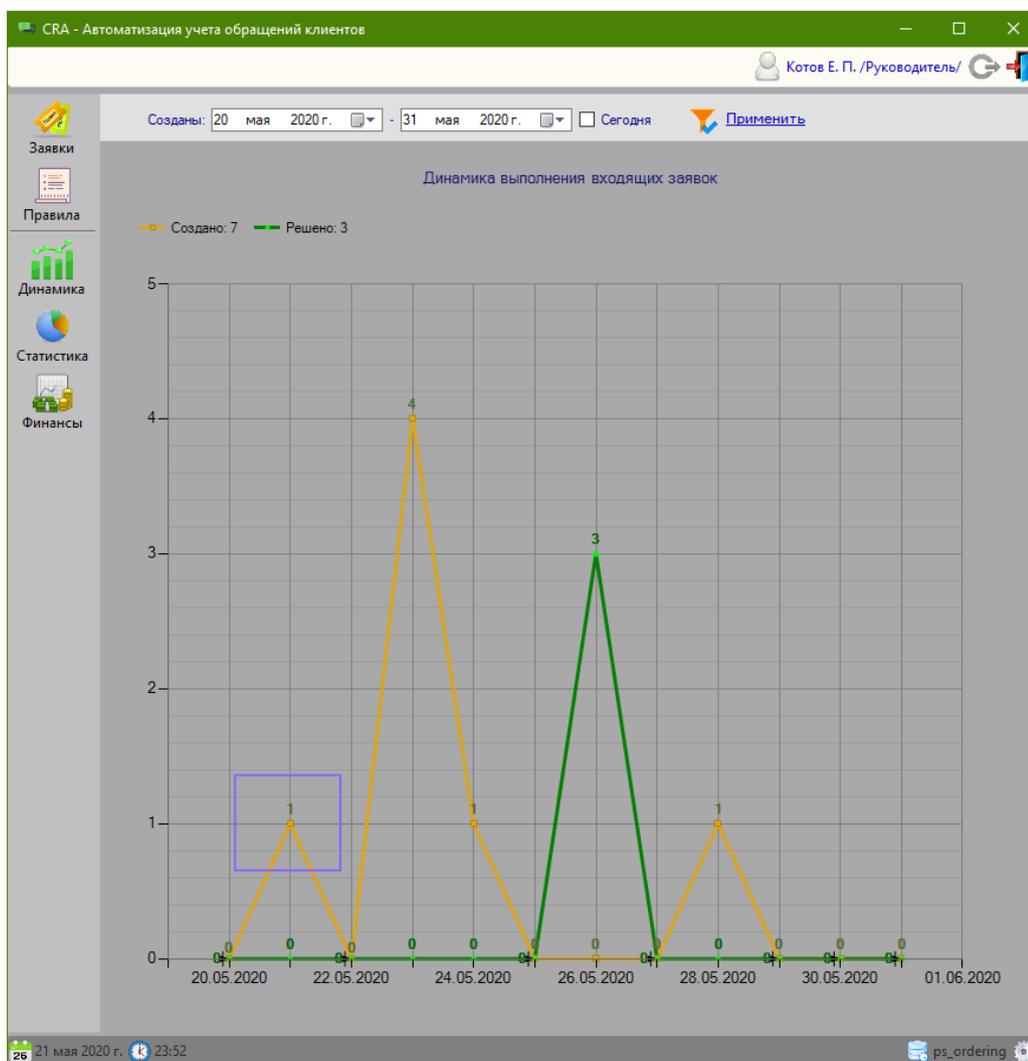


Рисунок 34 – Учет новой заявки в отчете по динамике обработки заявок

На рисунке 34 выделен фрагмент, соответствующий моменту создания новой заявки.

На рисунке 35 приведен результат построения сводного статистического отчета по результатам обработки заявок (на рисунке выделены фрагменты, соответствующие учету заявок).

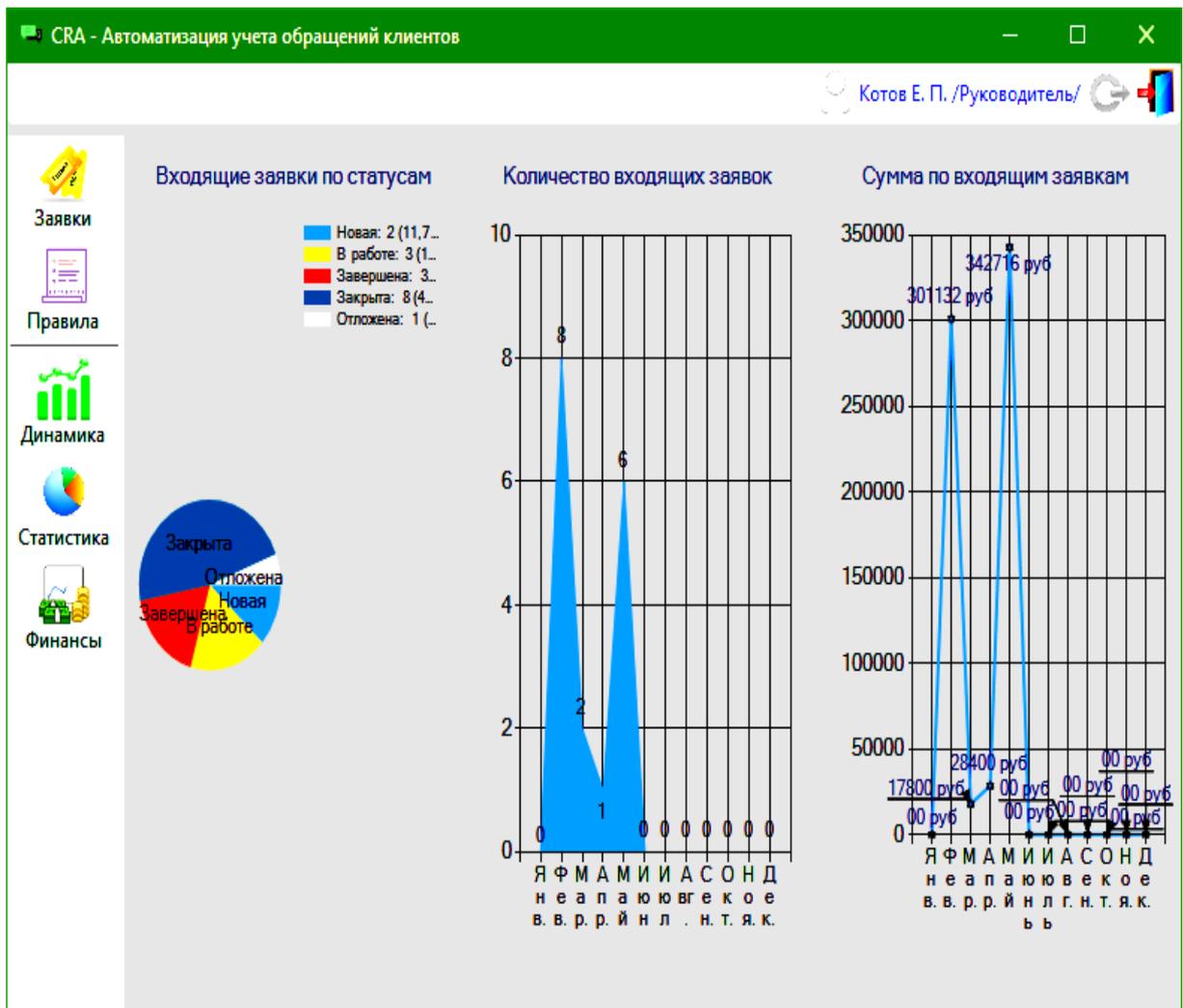


Рисунок 35 – Форма сводного статистического отчета

На рисунке 36 приведен результат построения сводного финансового отчета по результатам обработки заявок (на рисунке выделен фрагмент, соответствующий учету новой заявки).

CRA - Автоматизация учета обращений клиентов

Котов Е. П. /Руководитель/

Созданы: 1 мая 2020 г. - 31 мая 2020 г. Сегодня

Показать все

Сохранить файл... Просмотр документа Печать...

Отчет по входящим заявкам за период с 1 мая 2020 г. по 31 мая 2020 г.

| №п/п | № заявки | Тип заявки | Поступила | Запланировано | Решено | Суммарная стоимость обработки |
|------|----------|--|----------------|-------------------|----------------|-------------------------------|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 0002 | Установка СКУД | 23 мая 2020 г. | 23 июня 2020 г. | - | 0 Р |
| 2 | 0020 | Установка СКУД | 21 мая 2020 г. | 31 мая 2020 г. | - | 21 424 Р |
| 3 | 0019 | Разработка и внедрение информационных систем | 28 мая 2020 г. | 8 июня 2020 г. | - | 50 280 Р |
| 4 | 0007 | Разработка и внедрение информационных систем | 24 мая 2020 г. | 28 июня 2020 г. | - | 60 818 Р |
| 5 | 0001 | Интеграция систем анализа данных | 23 мая 2020 г. | 23 июня 2020 г. | 26 мая 2020 г. | 128 018 Р |
| 6 | 0004 | Проектирование информационных систем | 23 мая 2020 г. | 22 ноября 2020 г. | 26 мая 2020 г. | 103 600 Р |
| 7 | 0005 | Разработка и внедрение информационных систем | 23 мая 2020 г. | 23 мая 2021 г. | - | 0 Р |

Всего заявок обработано за указанный период: 7
из них решено: 2
Общая сумма по обработанным заявкам за указанный период: 364140

Руководитель отдела обработки заявок / _____ /

21 мая 2020 г.

Рисунок 36 – Учет новой заявки в финансовом отчете

Таким образом, видно, что данные тестовой заявки отображаются и учитываются во всех отчетах.

Выводы по главе 3

В ходе выполнения данного раздела была спроектирована и реализована АИС «Сибинтек: Управление заявками». Система была построена по модульному принципу, что соответствует хорошей архитектуре, позволяющей упростить понимание и дальнейшую поддержку, и развитие продукта.

Тестирование функций разработанной АИС показало, что система корректно выполняет свои функции.

Заключение

В ходе выполнения данной выпускной работы были рассмотрены и проанализированы бизнес-процессы ООО «ИК «Сибинтек» по учету и обработке входящих заявок обращений клиентов.

В ходе работы была разработана общая концепция организации процессов, сформулирована роль информационных систем в оптимизации процессов.

В работе приведена краткая характеристика существующих на рынке программного обеспечения программных продуктов, так или иначе способствующих решению выделенных задач, рассмотрены их возможности и недостатки. Также приведено описание программных средств, инструментов и технологий, с помощью которых можно разработать проект информационной системы и осуществить ее программную реализацию.

В процессе решения поставленных задач в рамках настоящего проекта были достигнуты следующие результаты:

- произведен анализ предметной области;
- разработана подробная модель предметной области;
- выполнено проектирование АИС «Сибинтек: Управление заявками»;
- разработан проект информационного обеспечения АИС «Сибинтек: Управление заявками»;
- выполнена реализация программного обеспечения АИС «Сибинтек: Управление заявками».

Результаты тестирования АИС «Сибинтек: Управление заявками» показали, что разработанная система полностью соответствует назначению и способна выполнять поставленные задачи. Таким образом, применение АИС «Сибинтек: Управление заявками» способно обеспечить повышение эффективности учета обработки входящих заявок обращений клиентов в ООО «ИК «Сибинтек».

Список используемой литературы

1. Агапов, О. Основные принципы работы CRM /О. Агапов, Д. Спиридонов. - М.: Издательство Граф, 2010. – 127с.
2. Албахари, Джозеф С#. 6.0 Справочник. Полное описание языка / Джозеф Албахари. - М: пер. с англ. ООО «И.Д. Вильямс», 2016. - 1040 с.
3. Арлоу, Д. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование: пер. с англ. / Д. Арлоу, И. Нейштадт. - Символ-Плюс, 2012. - 624 с.
4. Базы данных и экспертные системы [Текст]: учеб. пособие для студ. вузов / под. ред. Л.В. Логанова. - М.: Самара: Самарский национальный исследовательский университет имени академика С.П. Королева, 2011. – 12 с.
5. Балдин, К.В. Информационные системы в экономике. / К.В. Балдин, В.Б. Уткин. - М.: Издательский центр Академия, 2005. - 224 с.
6. Бардина, Н.В. Диаграмма классов и модель «сущность-связь» как логические модели информационной системы // Модели, системы, сети в экономике, технике, природе и обществе. - 2012. - № 2.- 92-95 с.
7. Вершинин, М. С# Enterprise Edition. Технологии проектирования и разработки. / М. Вершинин, Е. Иванова. - М.: ВHV, 2003. – 1088 с.
8. Головчинер, М.Н. Проектирование информационных систем. Методические указания по подготовке к государственному экзамену. / М.Н. Головчинер. - Томск, 2009. - 110 с.
9. Дейт, К.Дж. Введение в системы баз данных / К.Дж. Дейт. - М. Издательство Вильямс, пер. с англ., 2005. - 328 с.
10. Иванов, К.К. Использование функциональных зависимостей и нормализации при проектировании баз данных // Молодой ученый. - №19. - 26-27 с.
11. Линейная организационная структура: сайт. - URL: http://www.aup.ru/books/m151/1_5.htm (дата обращения: 07.03.2020). Текст: электронный.

12. Маклаков, С.В. ERWin и BPWin. CASE-средства разработки информационных систем / С.В. Маклаков. - М.: Москва, 1999. - 52 с.
13. Малыхина, М.П. Базы данных. Основы, проектирование, использование / М.П. Малыхина. – М.: С - Петербург, 2013. – 518 с.
14. Нотация BPMN: [Электронный ресурс]. - URL: <http://bpmssoft.org/bpmn> (дата обращения: 14.03.2020).
15. О компании ООО «ИК «Сибинтек»: [Электронный ресурс]. - URL: <https://sibintek.ru/company> (дата обращения: 30.03.2020).
16. О принятии и введении в действие рекомендаций по стандартизации: Постановление Госстандарта РФ от 02.07.2001 N 256-ст. –1с.
17. Обзор рынка CRM за 2019 год и направления его развития. Портал «Практика CRM» / В. Гусева. URL: <http://www.crm-practice.ru/editor/5849> (дата обращения: 20.03.2020).
18. Основы проектирования информационных систем [Текст]: учеб. пособие для студ. вузов / под. ред. И.Ю. Коцюба, А.В. Чунакова, А.Н. Шишкова. – М.: СПб: Университет ИТМО, 2015. - 206 с.
19. Основы формальных методов описания бизнес-процессов [Текст]: учеб. пособие для студ. вузов / под. ред.: К.Е. Самуйлова, Н.В. Серебренникова, А.В. Чукарина, Н.В. Яркина. - М.: РУДН, 2008. – 132 с.
20. Советов, Б.Я. Базы данных. Теория и практика. Учебник / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской. – М.: СПб: Высшая школа, 2015. – 420 с.
21. Тюгашев, А.А. Основы программирования. Часть I / А.А. Тюгашев. – М.: СПб: Университет ИТМО, 2016. – 285 с.
22. Эмблер, Скотт. Рефакторинг баз данных. Эволюционное проектирование. Книга / Эмблер Скотт, Прамодкумар Дж.Садаладж. - М.: Вильямс, 2010. - 18 с.
23. URL:http://sparxsystems.com/enterprise_architect_user_guide/14.0/index/index.html, (дата обращения 20.05.2020).

24. URL: <http://Object Management Group Inc.>, UML 2.0 Infrastructure Specification: Document number 03-09-15 (Спецификация инфраструктуры языка UML 2.0: Документ номер 03-09-15), (дата обращения 12.05.2020).

25. URL:[http:// Orientsoft. Software development & Business consulting.](http://Orientsoft. Software development & Business consulting.) Функциональное моделирование на базе IDEF0 (учебный курс). М.: Минск, 2002. – 22 с.

Приложение А

Скрипт формирования физической структуры базы данных АИС «Сибинтек: Управление заявками» на языке описания данных DDL (Data Description Language)

```
-- -----  
-- Generated by Enterprise Architect Version 10.0.1009  
-- Created On : понедельник, 18 Май, 2020  
-- DBMS      : MySql  
-- -----  
  
-- Create Tables  
CREATE TABLE Client  
(  
    cli_id INTEGER NOT NULL AUTO_INCREMENT,  
    cli_orgname VARCHAR(50) NOT NULL,  
    cli_furname VARCHAR(20) NOT NULL,  
    cli_firstname VARCHAR(20) NOT NULL,  
    cli_secondname VARCHAR(20),  
    cli_phone VARCHAR(20),  
    cli_email VARCHAR(20),  
    PRIMARY KEY (cli_id)  
);  
  
CREATE TABLE TimeRule  
(  
    dlr_id INTEGER NOT NULL AUTO_INCREMENT,  
    dlr_name VARCHAR(50),  
    dlr_id_type INTEGER,  
    dlr_registration INTEGER,  
    dlr_id_client INTEGER,  
    dlr_id_priority INTEGER,  
    dlr_normdays INTEGER NOT NULL,  
    PRIMARY KEY (dlr_id),  
    KEY (dlr_id_client),  
    KEY (dlr_id_type)  
);  
  
CREATE TABLE OrderType  
(  
    ort_id INTEGER NOT NULL AUTO_INCREMENT,  
    ort_name VARCHAR(100) NOT NULL,  
    PRIMARY KEY (ort_id)  
);  
  
CREATE TABLE RouteRule  
(  
    rtr_id INTEGER NOT NULL AUTO_INCREMENT,  
    rtr_name VARCHAR(50),  
    rtr_id_type INTEGER,  
    rtr_registration INTEGER,  
    rtr_id_client INTEGER,  
    rtr_id_priority INTEGER,  
    rtr_id_employee INTEGER,  
    PRIMARY KEY (rtr_id),  
    KEY (rtr_id_client),  
    KEY (rtr_id_employee),
```

Продолжение Приложения А

```
        KEY (rtr_id_type),
        KEY (rtr_id_priority)
    );
CREATE TABLE ApplyService
(
    aps_id_service INTEGER NOT NULL,
    aps_id_order INTEGER NOT NULL,
    aps_date DATE,
    aps_count INTEGER NOT NULL,
    aps_note VARCHAR(255),
    PRIMARY KEY (aps_id_service, aps_id_order),
    KEY (aps_id_order),
    KEY (aps_id_service)
);

CREATE TABLE Service
(
    srv_id INTEGER NOT NULL AUTO_INCREMENT,
    srv_name VARCHAR(100) NOT NULL,
    srv_price DECIMAL(10) NOT NULL,
    PRIMARY KEY (srv_id)
);

CREATE TABLE Priority
(
    pri_id INTEGER NOT NULL AUTO_INCREMENT,
    pri_name VARCHAR(10) NOT NULL,
    pri_level INTEGER NOT NULL,
    pri_color_r INTEGER NOT NULL,
    pri_color_g INTEGER NOT NULL,
    pri_color_b INTEGER NOT NULL,
    PRIMARY KEY (pri_id)
);

CREATE TABLE Status
(
    sts_id INTEGER NOT NULL AUTO_INCREMENT,
    sts_name VARCHAR(15) NOT NULL,
    sts_color_r INTEGER NOT NULL,
    sts_color_g INTEGER NOT NULL,
    sts_color_b INTEGER NOT NULL,
    PRIMARY KEY (sts_id)
);

CREATE TABLE Employee
(
    emp_id INTEGER NOT NULL AUTO_INCREMENT,
    emp_furname VARCHAR(20),
    emp_firstname VARCHAR(20),
    emp_secondname VARCHAR(20),
    emp_id_dep INTEGER NOT NULL,
    emp_login VARCHAR(6) NOT NULL,
    emp_password VARCHAR(6) NOT NULL,
    emp_account INTEGER NOT NULL,
    PRIMARY KEY (emp_id),
    KEY (emp_id_dep)
);
```

Продолжение Приложения А

```
CREATE TABLE Department
(
    dep_id INTEGER NOT NULL AUTO_INCREMENT,
    dep_name VARCHAR(100) NOT NULL,
    dep_phone VARCHAR(20),
    PRIMARY KEY (dep_id)
);
CREATE TABLE Comment
(
    com_id INTEGER NOT NULL AUTO_INCREMENT,
    com_id_order INTEGER NOT NULL,
    com_date DATETIME NOT NULL,
    com_text TEXT NOT NULL,
    com_image BLOB,
    com_id_employee INTEGER NOT NULL,
    PRIMARY KEY (com_id),
    KEY (com_id_employee),
    KEY (com_id_order)
);
CREATE TABLE Order
(
    ord_id INTEGER NOT NULL AUTO_INCREMENT,
    ord_registration INTEGER NOT NULL,
    ord_create DATETIME NOT NULL,
    ord_description TEXT,
    ord_id_client INTEGER NOT NULL,
    ord_id_priority INTEGER NOT NULL,
    ord_deadline DATE,
    ord_id_status INTEGER NOT NULL,
    ord_id_executor INTEGER NOT NULL,
    ord_complete DATE,
    ord_id_type INTEGER NOT NULL,
    PRIMARY KEY (ord_id),
    KEY (ord_id_client),
    KEY (ord_id_executor),
    KEY (ord_id_type),
    KEY (ord_id_priority),
    KEY (ord_id_status)
);
-- Create Foreign Key Constraints
ALTER TABLE TimeRule ADD CONSTRAINT FK_TimeRule_Client
    FOREIGN KEY (dlr_id_client) REFERENCES Client (cli_id)
    ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE TimeRule ADD CONSTRAINT FK_TimeRule_OrderType
    FOREIGN KEY (dlr_id_type) REFERENCES OrderType (ort_id)
    ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE RouteRule ADD CONSTRAINT FK_RouteRule_Client
    FOREIGN KEY (rtr_id_client) REFERENCES Client (cli_id)
    ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE RouteRule ADD CONSTRAINT FK_RouteRule_Employee
    FOREIGN KEY (rtr_id_employee) REFERENCES Employee (emp_id)
    ON DELETE CASCADE ON UPDATE CASCADE;
```

Продолжение Приложения А

```
ALTER TABLE RouteRule ADD CONSTRAINT FK_RouteRule_OrderType
FOREIGN KEY (rtr_id_type) REFERENCES OrderType (ort_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE RouteRule ADD CONSTRAINT FK_RouteRule_Priority
FOREIGN KEY (rtr_id_priority) REFERENCES Priority (pri_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE ApplyService ADD CONSTRAINT FK_ApplyService_Order
FOREIGN KEY (aps_id_order) REFERENCES Order (ord_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE ApplyService ADD CONSTRAINT FK_ApplyService_Service
FOREIGN KEY (aps_id_service) REFERENCES Service (srv_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE Employee ADD CONSTRAINT FK_Employee_Department
FOREIGN KEY (emp_id_dep) REFERENCES Department (dep_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE Comment ADD CONSTRAINT FK_Comment_Employee
FOREIGN KEY (com_id_employee) REFERENCES Employee (emp_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE Comment ADD CONSTRAINT FK_Comment_Order
FOREIGN KEY (com_id_order) REFERENCES Order (ord_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE Order ADD CONSTRAINT FK_Order_Client
FOREIGN KEY (ord_id_client) REFERENCES Client (cli_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE Order ADD CONSTRAINT FK_Order_Employee
FOREIGN KEY (ord_id_executor) REFERENCES Employee (emp_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE Order ADD CONSTRAINT FK_Order_OrderType
FOREIGN KEY (ord_id_type) REFERENCES OrderType (ort_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE Order ADD CONSTRAINT FK_Order_Priority
FOREIGN KEY (ord_id_priority) REFERENCES Priority (pri_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE Order ADD CONSTRAINT FK_Order_Status
FOREIGN KEY (ord_id_status) REFERENCES Status (sts_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

Приложение Б

Чертежи форм документов и видеок кадров АИС «Сибинтек: Управление заявками»

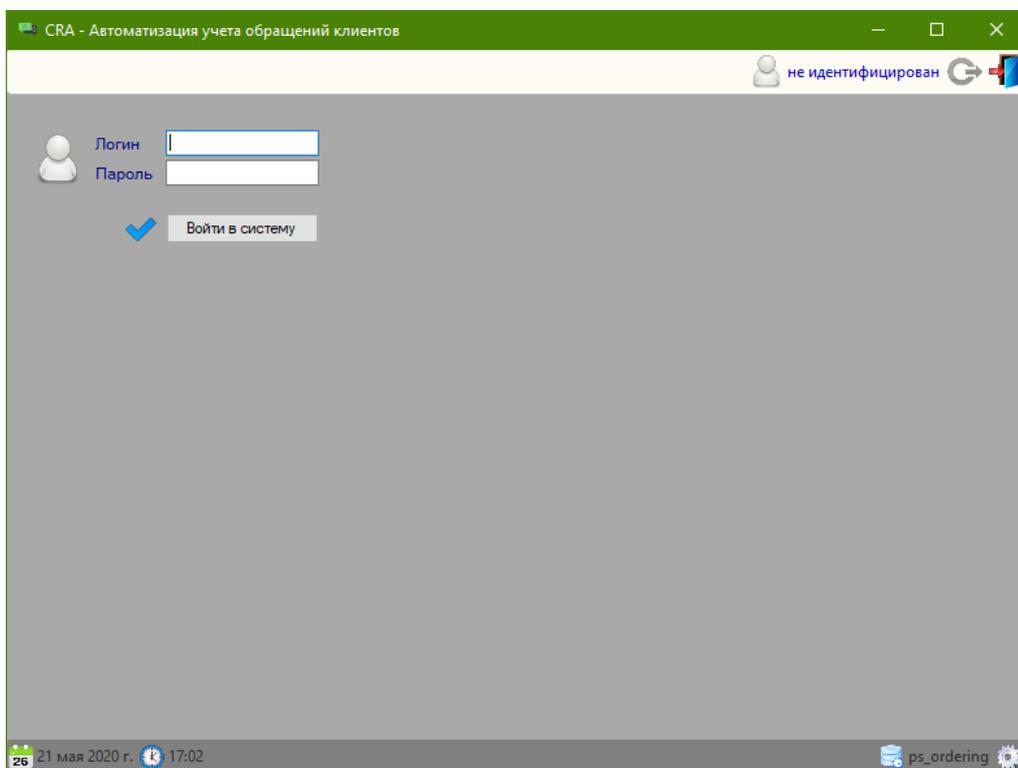


Рисунок Б.1 – Форма авторизации в системе

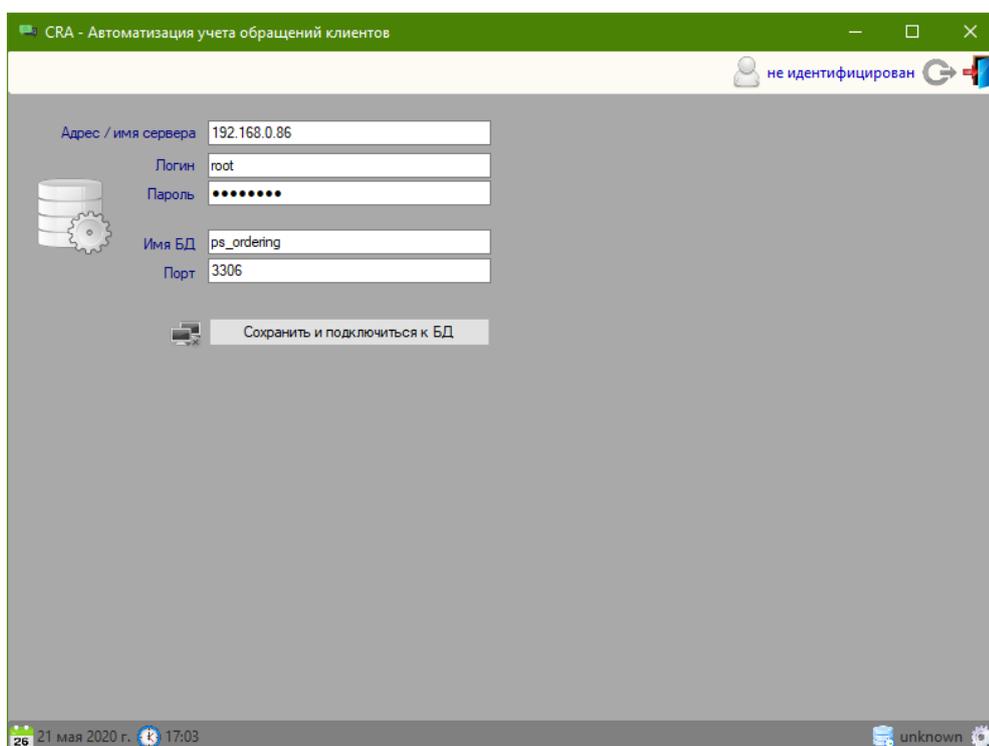


Рисунок Б.2 – Форма настройки подключения к БД

Продолжение Приложения Б

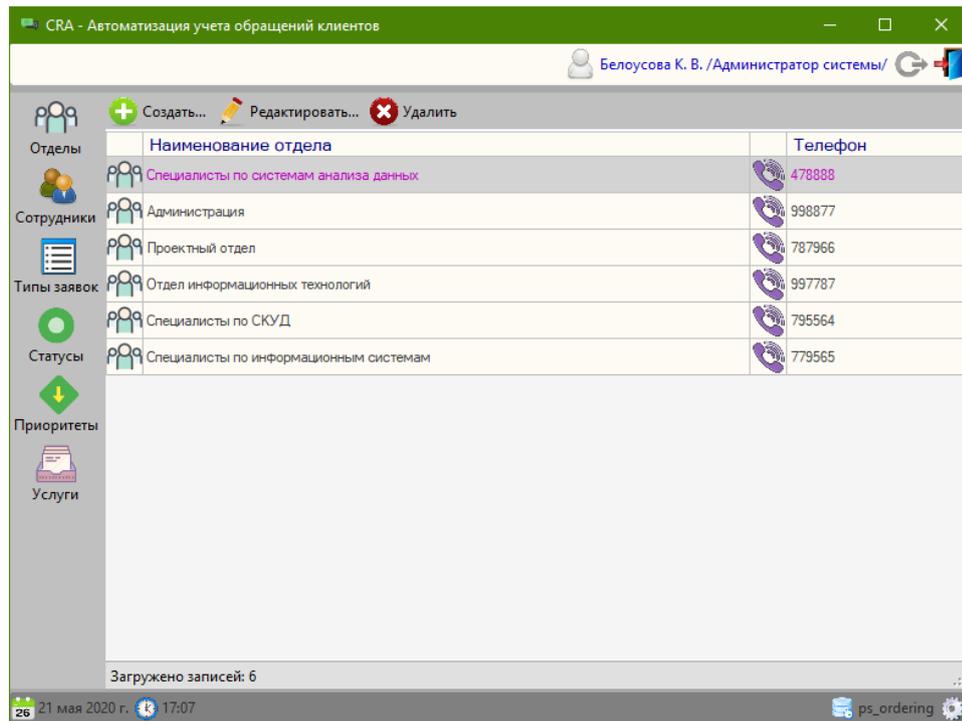


Рисунок Б.3 – Форма Вид подсистемы справочника подразделений

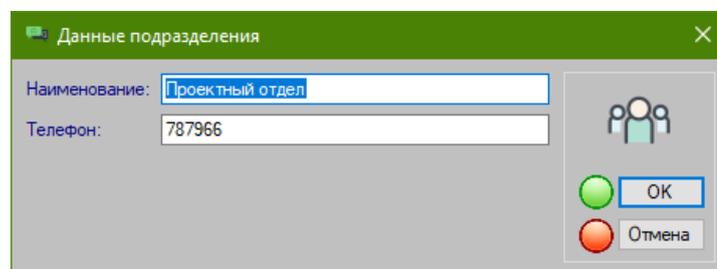
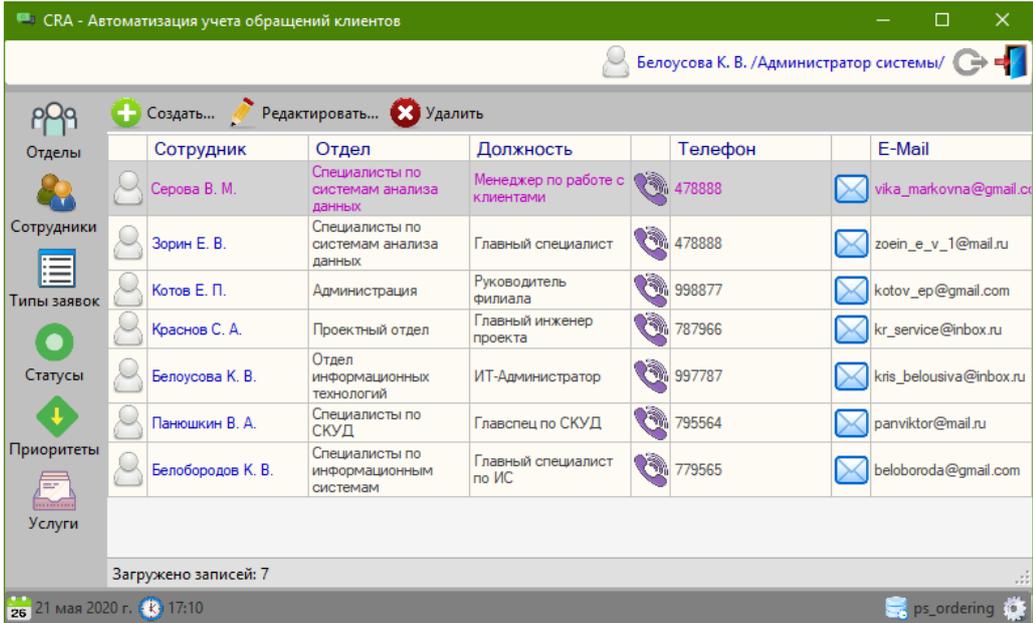


Рисунок Б.4 – Форма ввода данных подразделения

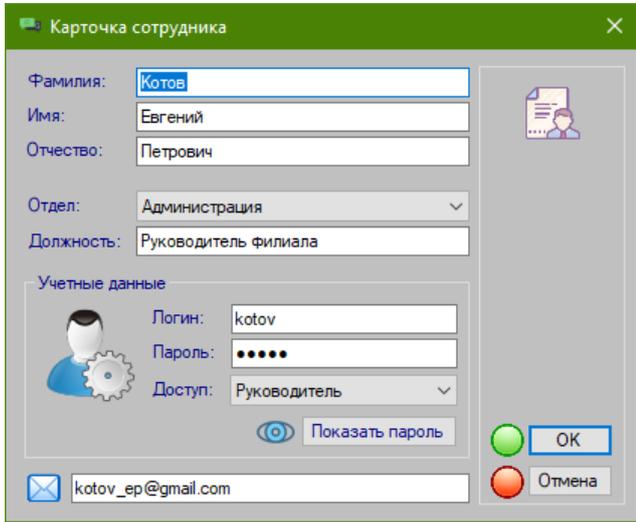
Продолжение Приложения Б



The screenshot shows the 'CRA - Автоматизация учета обращений клиентов' application. The user is logged in as 'Белюсова К. В. /Администратор системы/'. The main area displays a table of employees with columns for 'Сотрудник', 'Отдел', 'Должность', 'Телефон', and 'E-Mail'. The table contains 7 records. A sidebar on the left has navigation icons for 'Отделы', 'Сотрудники', 'Типы заявок', 'Статусы', 'Приоритеты', and 'Услуги'. At the bottom, it shows the date '21 мая 2020 г.' and time '17:10'.

| Сотрудник | Отдел | Должность | Телефон | E-Mail |
|-------------------|--|--------------------------------|---------|-------------------------|
| Серова В. М. | Специалисты по системам анализа данных | Менеджер по работе с клиентами | 478888 | vika_markovna@gmail.com |
| Зорин Е. В. | Специалисты по системам анализа данных | Главный специалист | 478888 | zoein_e_v_1@mail.ru |
| Котов Е. П. | Администрация | Руководитель филиала | 998877 | kotov_ep@gmail.com |
| Краснов С. А. | Проектный отдел | Главный инженер проекта | 787966 | kr_service@inbox.ru |
| Белюсова К. В. | Отдел информационных технологий | ИТ-Администратор | 997787 | kris_belousiva@inbox.ru |
| Панюшкин В. А. | Специалисты по СКУД | Главспец по СКУД | 795564 | panviktor@mail.ru |
| Белобородов К. В. | Специалисты по информационным системам | Главный специалист по ИС | 779565 | beloboroda@gmail.com |

Рисунок Б.5 – Форма справочника сотрудников



The screenshot shows the 'Карточка сотрудника' form. It contains the following fields and values:

- Фамилия: Котов
- Имя: Евгений
- Отчество: Петрович
- Отдел: Администрация
- Должность: Руководитель филиала
- Учетные данные:
 - Логин: kotov
 - Пароль: [masked]
 - Доступ: Руководитель
- Additional email: kotov_ep@gmail.com

Buttons: 'Показать пароль', 'ОК', 'Отмена'.

Рисунок Б.6 – Форма ввода данных сотрудника

Продолжение Приложения Б

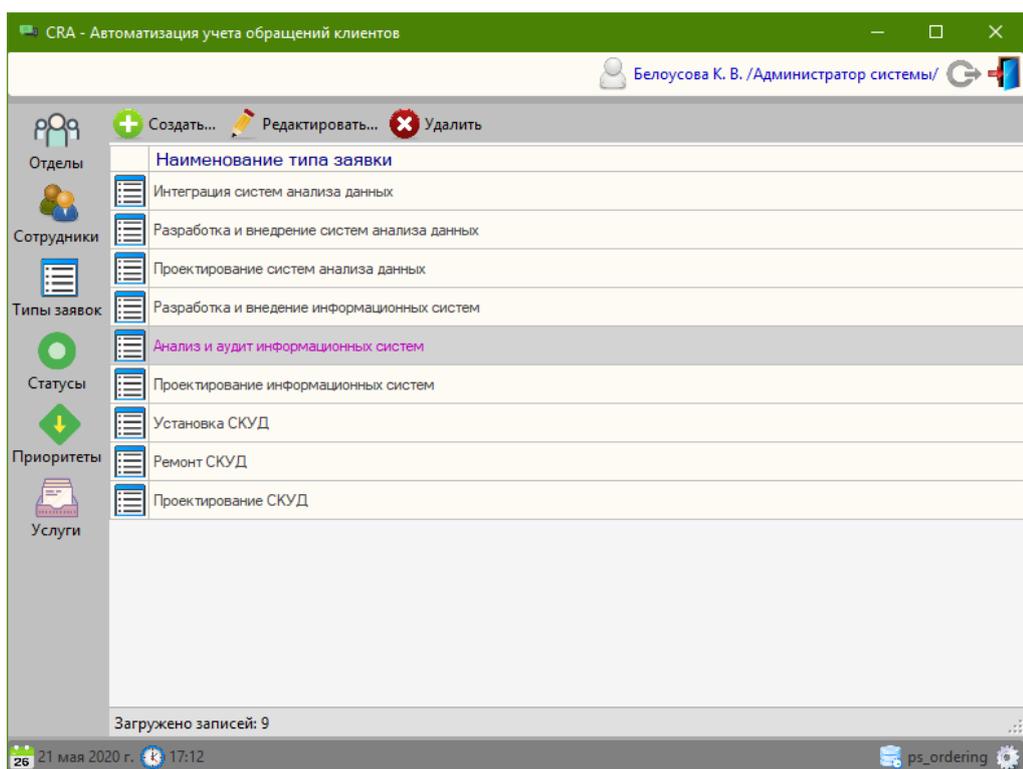


Рисунок Б.7 – Форма справочника типов заявок

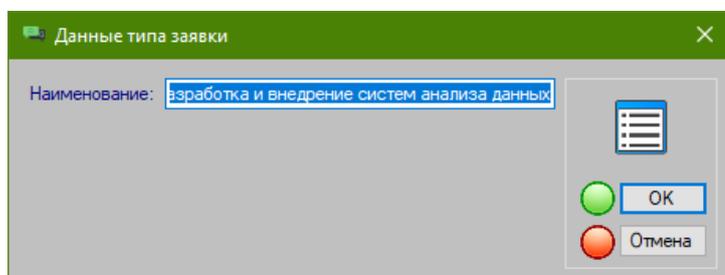


Рисунок Б.8 – Форма ввода данных типа заявки

Продолжение Приложения Б

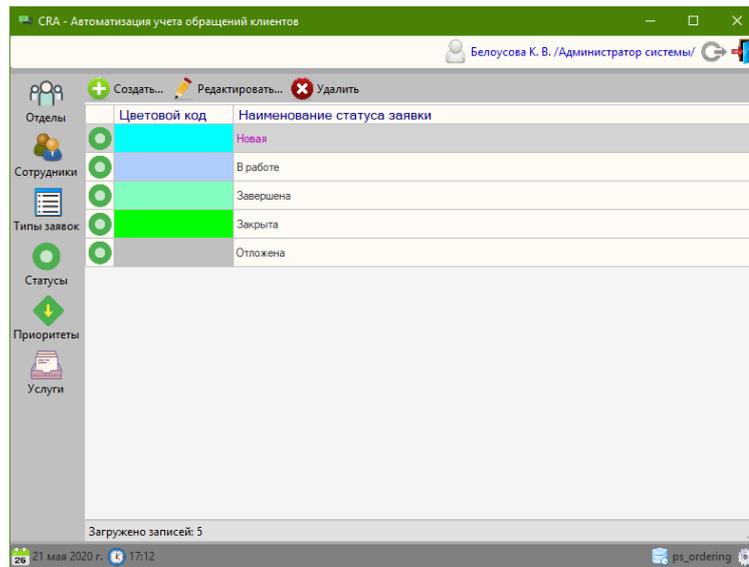


Рисунок Б.9 – Форма справочника статусов заявок

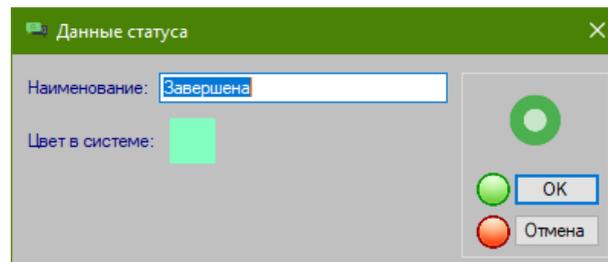


Рисунок Б.10 – Форма ввода данных статуса заявки

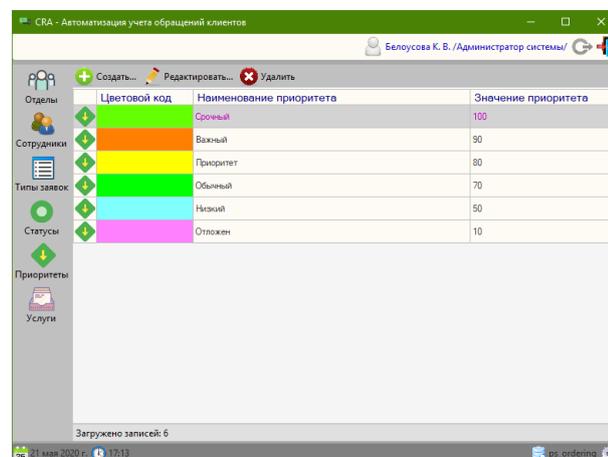


Рисунок Б.11 – Форма справочника приоритетов заявок

Продолжение Приложения Б

Данные приоритета

Наименование: Обычный

Уровень: 70

Цвет в системе: ■

OK

Отмена

Рисунок Б.12 – Форма ввода данных приоритета заявки

CRA - Автоматизация учета обращений клиентов

Белусова К. В. /Администратор системы/

Создать... Редактировать... Удалить

| Наименование работы (услуги) | Стоимость |
|---|-----------|
| Монтаж датацентра | 1 500 Р |
| Установка серверного шкафа | 1 500 Р |
| Прокладка кабеля (наружная прокладка), м. | 100 Р |
| Монтаж и подключение блока питания | 750 Р |
| Переходник DMA | 4 206 Р |
| Жесткий диск 500 ГБ | 4 200 Р |
| Монитор 19" | 8 100 Р |
| Система охлаждения сетевого шкафа | 8 800 Р |
| Блок питания БП 12/2А | 712 Р |
| Разъем BNC | 280 Р |
| Модуль памяти | 3 500 Р |
| Аккумулятор 12В 7Ач | 1 000 Р |
| CD-диск | 180 Р |
| Монтаж КТС информационных систем | 14 500 Р |

Загружено записей: 33

26 21 мая 2020 г. 17:14 ps_ordering

Рисунок Б.13 – Форма справочника услуг (работ)

Продолжение Приложения Б

Данные услуги

Наименование: Система охлаждения сетевого шкафа

Стоимость, руб.: 8800

OK

Отмена

Рисунок Б.14 – Форма ввода данных услуги (работы)

Выбор услуг / работ

Подтвердить выбор

| Наименование работы (услуги) | Стоимость |
|---|-----------|
| Монтаж датацентра | 1 500 Р |
| Установка серверного шкафа | 1 500 Р |
| Прокладка кабеля (наружная прокладка), м. | 100 Р |
| Монтаж и подключение блока питания | 750 Р |
| Переходник DMA | 4 206 Р |
| Жесткий диск 500 ГБ | 4 200 Р |
| Монитор 19" | 8 100 Р |
| Система охлаждения сетевого шкафа | 8 800 Р |
| Блок питания БП 12/2А | 712 Р |

Комментарий:

Рисунок Б.15 – Форма выбора услуги (работы)

CRA - Автоматизация учета обращений клиентов

Серова В. М. / Менеджер по клиентам

Создать... Редактировать... Удалить

| Заявки | Организация | Фамилия | Имя | Отчество | Телефон |
|------------|-------------------------------|-------------|------------|------------|----------------------|
| Клиенты | ООО "Стальные системы" | Соколовский | Виталий | Евгеньевич | 89001110022 |
| Клиенты | ООО "Полиграфические системы" | Марьина | Светлана | Павловна | 89000123311 |
| Клиенты | ОО "Деловая клякса" | Сотникова | Екатерина | Олеговна | 89099099090 |
| Клиенты | ООО "Феникс Контакт Рус" | Рыжик | Евгений | Васильевич | 8-495-933-8548 д.182 |
| Клиенты | ООО "Риттал, Воронеж" | Безуглый | Яков | Николаевич | +7(473)234 07 54 |
| Сотрудники | ИП Максим | Аверьянов | Максим | Максимович | 89001110101 |
| Сотрудники | АО "СОФТ-ИНЖИНИРИНГ" | Попова | Александра | Витальевна | 89110002200 |
| Сотрудники | Частный дом | Бромов | Сергей | Иванович | 89001112211 |
| Сотрудники | ИП Ольга | Саввина | Ольга | Артемовна | 91110002231 |
| Сотрудники | ИП Романов | Романов | Петр | Васильевич | 80001113322 |
| Сотрудники | АО "Барокко" | Лосева | Инга | Андреевна | 89001112211 |
| Сотрудники | ДВИН | Ростовский | Илья | Ильич | 89774441155 |
| Сотрудники | АО "БОСТОН" | Иридьев | Виталий | Андреевич | 89001112121 |
| Сотрудники | ООО "Савой" | Пронина | Марина | Николаевна | 89001112211 |

Загружено записей: 14

26 21 мая 2020 г. 17:17 ps_ordering

Рисунок Б.16 – Форма справочника клиентов

Продолжение Приложения Б

Данные клиента

Наименование: ИП Максим

Фамилия: Аверьянов

Имя: Максим

Отчество: Максимович

89001110101

avmaxmax@ipmax.com

OK

Отмена

Рисунок Б.17 – Форма ввода данных клиента

CRA - Автоматизация учета обращений клиентов

Серова В. М. /Менеджер по клиентам/

Правила маршрутизации заявок | Правила срока выполнения заявок

Заявки: Создать... | Редактировать... | Удалить

| Имя правила | Тип заявки | Способ регистрации | Клиент | Приоритет | Ответственный |
|--|--|--------------------|--------|-----------|-------------------|
| Установка СКУД | Установка СКУД | Любой | Любой | Любой | Панюшкин В. А. |
| Интеграция систем анализа данных | Интеграция систем анализа данных | Любой | Любой | Любой | Зорин Е. В. |
| Разработка и внедрение ИС | Разработка и внедрение информационных систем | Любой | Любой | Любой | Белобородов К. В. |
| Ремонт СКУД | Ремонт СКУД | Любой | Любой | Любой | Панюшкин В. А. |
| Разработка и внедрение систем анализа данных | Разработка и внедрение систем анализа данных | Любой | Любой | Любой | Зорин Е. В. |
| Анализ и аудит ИС | Анализ и аудит информационных систем | Любой | Любой | Любой | Белобородов К. В. |
| Проектирование СКУД | Проектирование СКУД | Любой | Любой | Любой | Краснов С. А. |
| Проектирование систем анализа данных | Проектирование систем анализа данных | Любой | Любой | Любой | Краснов С. А. |
| Проектирование ИС | Проектирование информационных систем | Любой | Любой | Любой | Краснов С. А. |

Загружено записей: 9

26 | 21 мая 2020 г. | 17:17 | ps_ordering

Рисунок Б.18 – Форма справочника правил маршрутизации входящих заявок

Продолжение Приложения Б

CRA - Автоматизация учета обращений клиентов

Серова В. М. / Менеджер по клиентам

Правила маршрутизации заявок Правила срока выполнения заявок

Заявки Создать... Редактировать... Удалить

| Имя правила | Тип заявки | Способ регистрации | Клиент | Проритет | Срок, дн. |
|-------------------------------------|--|--------------------|--------|-----------|-----------|
| Приоритетные | Любой | Любой | Любой | Приоритет | 30 |
| Срочные: интеграция САД | Интеграция систем анализа данных | Любой | Любой | Срочный | 15 |
| Срочные: Разработка и внедрение САД | Разработка и внедрение систем анализа данных | Любой | Любой | Срочный | 5 |
| Срочные: проектирование САД | Проектирование систем анализа данных | Любой | Любой | Срочный | 35 |
| Срочные: установка и внедрение ИС | Разработка и внедрение информационных систем | Любой | Любой | Срочный | 10 |
| Срочные: анализ и аудит ИС | Анализ и аудит информационных систем | Любой | Любой | Срочный | 6 |
| Срочные: проектирование ИС | Проектирование информационных систем | Любой | Любой | Срочный | 30 |
| Срочные: установка СКУД | Установка СКУД | Любой | Любой | Срочный | 10 |
| Срочные: ремонт СисПТ | Ремонт СКУД | Любой | Любой | Срочный | 5 |
| Срочные: проектирование СисПТ | Проектирование СКУД | Любой | Любой | Срочный | 45 |
| С низким приоритетом | Любой | Любой | Любой | Низкий | 180 |

Загружено записей: 12

26 21 мая 2020 г. 17:18 ps_ordering

Рисунок Б.19 – Форма справочника правил определения дедлайна для решения входящих заявок

Выбор клиента

Подтвердить выбор

| Организация | Фамилия | Имя | Отчество | Телефон |
|-------------------------------|-------------|------------|------------|----------------------|
| ООО "Стальные системы" | Соколовский | Виталий | Евгеньевич | 89001110022 |
| ООО "Полиграфические системы" | Марьина | Светлана | Павловна | 89000123311 |
| ООО "Деловая клякса" | Сотникова | Екатерина | Олеговна | 89099099090 |
| ООО "Феникс Контакт Рус" | Рыжих | Евгений | Васильевич | 8-495-933-8548 д.182 |
| ООО "Риттал, Воронеж" | Безуглый | Яков | Николаевич | +7(473)234 07 54 |
| ИП Максим | Аверьянов | Максим | Максимович | 89001110101 |
| АО "СОФТ-ИНЖИНИРИНГ" | Попова | Александра | Витальевна | 89110002200 |

Рисунок Б.20 – Форма выбора клиента

Продолжение Приложения Б

Диалоговое окно с заголовком «Данные правила маршрутизации заявки». В поле «Наименование правила» введено «Интеграция систем анализа данных». В разделе «ЕСПИ:» (Есть ли следующие параметры) заданы: Тип заявки = «Интеграция систем анализа данных» (выпадающий список), Любая (не отмечено); Сп-б регистрации = (выпадающий список), Любая (отмечено); Приоритет = (выпадающий список), Любая (отмечено); Клиент = (выпадающий список), Любая (отмечено). В разделе «ТО:» (Техническое описание) задан Ответственный = «Зорин Е. В.». В правом нижнем углу расположены кнопки «OK» (зеленая) и «Отмена» (красная).

Рисунок Б.21 – Форма ввода данных правила маршрутизации входящих заявок

Диалоговое окно с заголовком «Данные правила срока исполнения заявки». В поле «Наименование правила» введено «Срочные: анализ и аудит ИС». В разделе «ЕСПИ:» (Есть ли следующие параметры) заданы: Тип заявки = «Анализ и аудит информационных систем» (выпадающий список), Любая (не отмечено); Сп-б регистрации = (выпадающий список), Любая (отмечено); Приоритет = «Срочный» (выпадающий список), Любая (не отмечено); Клиент = (выпадающий список), Любая (отмечено). В разделе «ТО:» (Техническое описание) задан Срок исполнения, дн. = «6». В правом нижнем углу расположены кнопки «OK» (зеленая) и «Отмена» (красная).

Рисунок Б.22 – Форма ввода данных правила определения дедлайна для решения входящих заявок

Продолжение Приложения Б

Новая заявка

Клиент: ООО "Полиграфические системы"

Выбрать... Новый...

Тип: Интеграция систем анализа данных

Приоритет: Срочный

Описание:

Правило обработки срока: Срочные: интеграция САД
Правило обработки исполнителя: Интеграция систем анализа данных

Обработать до: 5 июня 2020 г.

Ответственный: Зорин Е. В.

Тикет

OK

Отмена

Рисунок Б.23 – Форма ввода данных новой входящей заявки

Продолжение Приложения Б

Краснов С. А. /Мастер-Исполнитель/

Создать... Удалить Просмотр заявки... Применить цвет статуса...

| ID | Тип | Создана | Закрыта | Клиент | Статус | Решить | Тип регистр | Ответствен |
|------|--------------------------------------|------------|------------|-------------------------------|-----------|------------|------------------|---------------|
| 0015 | Проектирование информационных систем | 22.02.2020 | | АО "БОСТОН" | В работе | 18.08.2020 | Служба менеджера | Краснов С. А. |
| 0017 | Проектирование информационных систем | 12.03.2020 | | ООО "Стальные системы" | В работе | 12.03.2020 | Служба менеджера | Краснов С. А. |
| 0016 | Проектирование систем анализа данных | 22.02.2020 | | ООО "Савой" | Завершена | 18.08.2020 | Служба менеджера | Краснов С. А. |
| 0001 | Интеграция систем анализа данных | 23.05.2020 | 26.05.2020 | ООО "Полиграфические системы" | Закрыта | 23.06.2020 | Служба менеджера | Краснов С. А. |
| 0004 | Проектирование информационных систем | 23.05.2020 | 26.05.2020 | АО "СОФТ-ИНЖИНИРИНГ" | Закрыта | 22.11.2020 | Служба менеджера | Краснов С. А. |
| 0013 | Проектирование СКУД | 09.02.2020 | 24.03.2020 | АО "Барокко" | Закрыта | 08.03.2020 | Служба менеджера | Краснов С. А. |

Клиент: Применить фильтр
 Приоритет: Сбросить фильтр
 Тип:
 Созданы: 1 января 2020 г. - 31 мая 2020 г. Сегодня
 Показать завершенные

Загружено записей: 6 Не закрытых заявок: 3 Решить сегодня: 0

21 мая 2020 г. 17:15 ps_ordering

Рисунок Б.24 – Форма учета входящих заявок

Продолжение Приложения Б

Просмотр заявки
✕

Заявка №0004

Создана: 23 мая 2020 г.
Обработать до: 22 ноября 2020 г.
Закрыта: 26 мая 2020 г.

Клиент: АО "СОФТ-ИНЖИНИРИНГ"
Попова А. В., т.: 89110002200; mail: alporova@engineering

Тип заявки: Проектирование информационных систем
Приоритет заявки: Низкий

Ответственный: Краснов С. А., т.: 787966
Проектный отдел ; Главный инженер проекта

Описание:

Комментарии + [Добавить](#)

Краснов С. А. 24.05.2020 15:41

Принял

Краснов С. А. 25.05.2020 10:54

Полный комплект тех. документации отправлен заказчику.

Серова В. М. 26.05.2020 10:55

ОК. Закрыто.

Выполненные работы / учтенные средства

+ [Добавить](#)
✕ [Удалить](#)
💬 [Описание](#)
💾 [Сохранить](#)

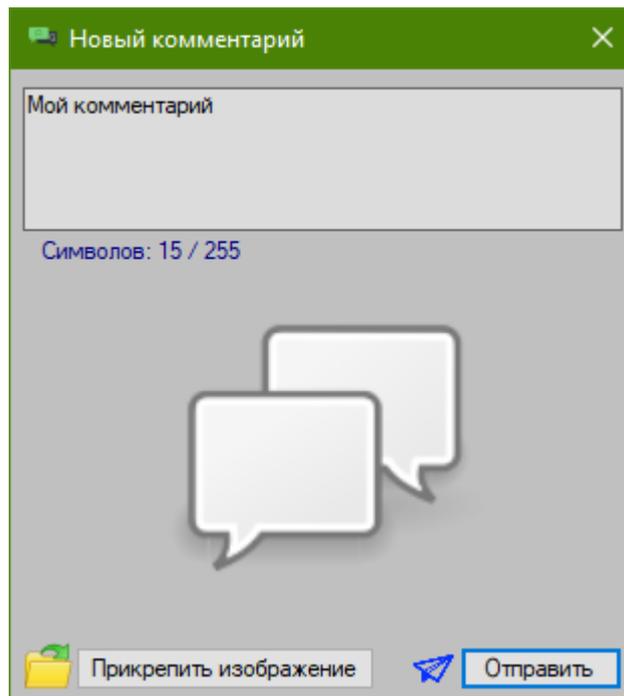
| | Дата | Работа / примененное оборудование | Стоим. за единицу | Кол-во | Сумма |
|--|------------|-----------------------------------|-------------------|--------|----------|
| | 25.05.2020 | Проектные работы. Раздел ЗМ | 8 000 Р | 4 | 32 000 Р |
| | 25.05.2020 | Проектные работы. Раздел АС | 9 000 Р | 4 | 36 000 Р |
| | 25.05.2020 | Проектные работы. Раздел СВН | 5 000 Р | 4 | 20 000 Р |
| | 25.05.2020 | Проектные работы. Раздел ОР | 3 900 Р | 4 | 15 600 Р |

Записей загружено: 4 Всего на сумму: 103 600 Р

Сменить статус: Закрыта
✔ [OK](#)

Рисунок Б.25 – Форма просмотра заявки

Продолжение Приложения Б



Новый комментарий

Мой комментарий

Символов: 15 / 255

Прикрепить изображение

Отправить

Рисунок Б.26 – Форма ввода комментария по заявке

Продолжение Приложения Б

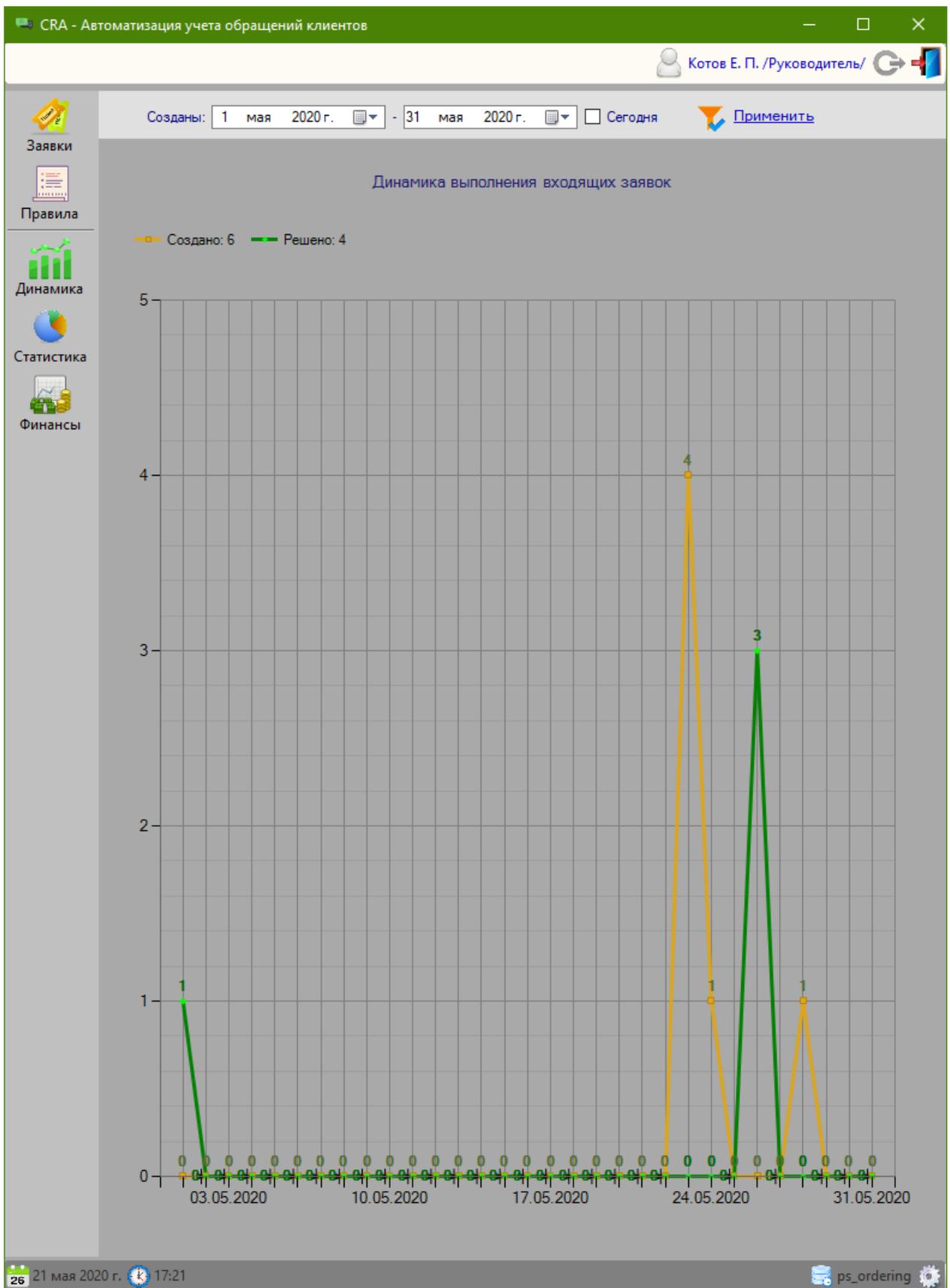


Рисунок Б.27 – Форма сводного отчета по динамике обработки заявок

Продолжение Приложения Б

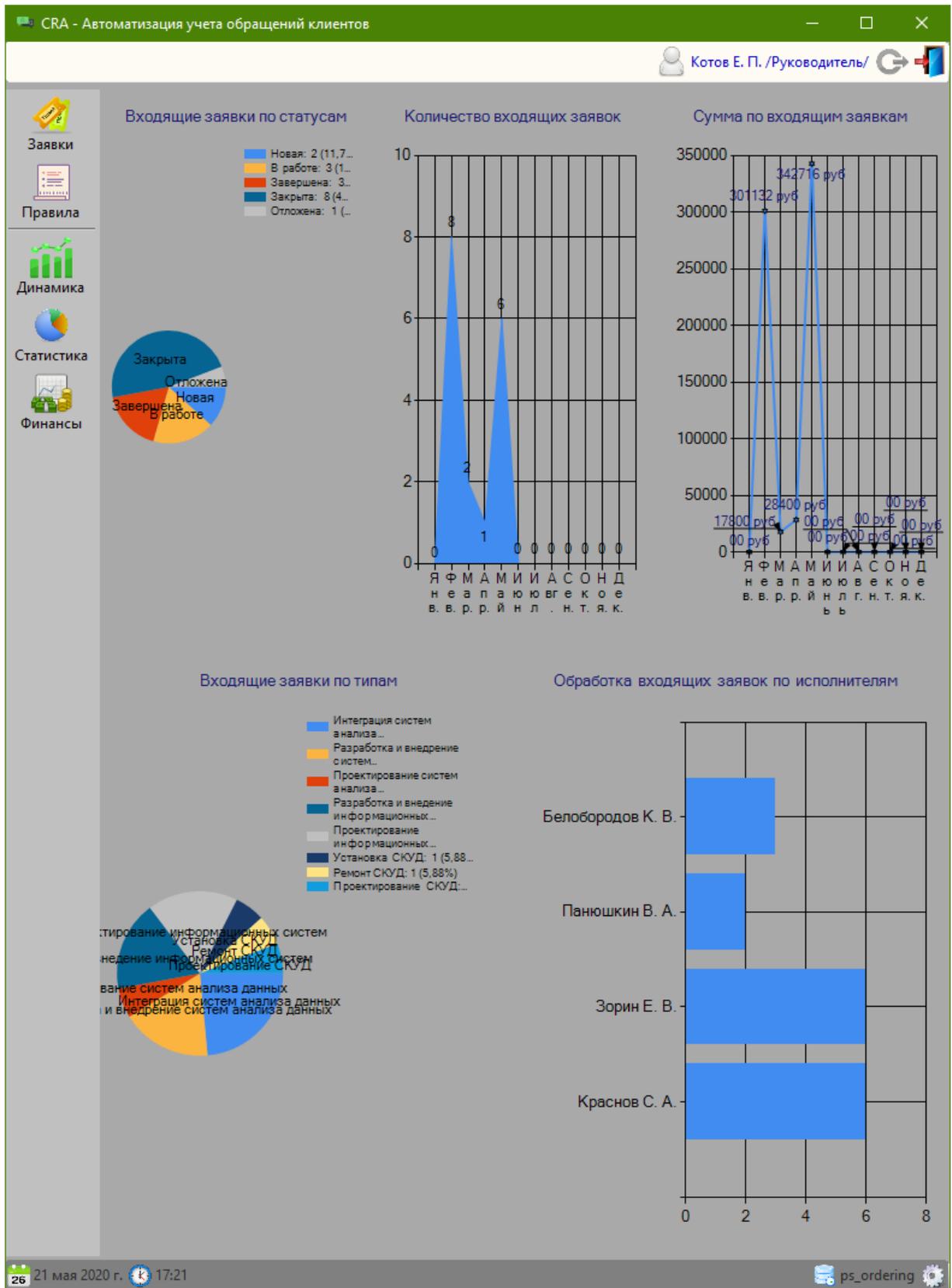


Рисунок Б.28 – Форма сводного статистического отчета

Продолжение Приложения Б

Котов Е. П. /Руководитель/

Созданы: 1 января 2020 г. - 31 мая 2020 г. Сегодня [Применить](#)

Показать все

[Сохранить файл...](#) [Просмотр документа](#) [Печать...](#)

Отчет по входящим заявкам за период с 1 января 2020 г. по 31 мая 2020 г.

| №п/п | № заявки | Тип заявки | Поступила | Запланировано | Решено | Суммарная стоимость обработки |
|------|----------|--|--------------------|--------------------|------------------|-------------------------------|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 0001 | Интеграция систем анализа данных | 23 мая 2020 г. | 23 июня 2020 г. | 26 мая 2020 г. | 128 018 Р |
| 2 | 0004 | Проектирование информационных систем | 23 мая 2020 г. | 22 ноября 2020 г. | 26 мая 2020 г. | 103 600 Р |
| 3 | 0008 | Разработка и внедрение систем анализа данных | 11 февраля 2020 г. | 26 апреля 2020 г. | 17 марта 2020 г. | 56 870 Р |
| 4 | 0009 | Разработка и внедрение систем анализа данных | 11 апреля 2020 г. | 29 мая 2020 г. | 1 мая 2020 г. | 28 400 Р |
| 5 | 0010 | Интеграция систем анализа данных | 9 февраля 2020 г. | 9 февраля 2020 г. | 24 марта 2020 г. | 12 300 Р |
| 6 | 0012 | Интеграция систем анализа данных | 9 февраля 2020 г. | 9 февраля 2020 г. | 12 марта 2020 г. | 55 216 Р |
| 7 | 0013 | Проектирование СКУД | 9 февраля 2020 г. | 8 марта 2020 г. | 24 марта 2020 г. | 33 900 Р |
| 8 | 0014 | Интеграция систем анализа данных | 11 февраля 2020 г. | 24 февраля 2020 г. | 26 мая 2020 г. | 114 986 Р |

Всего заявок обработано за указанный период: 8
из них решено: 8
Общая сумма по обработанным заявкам за указанный период: 533290

Руководитель отдела обработки заявок / _____ /

21 мая 2020 г.

26 21 мая 2020 г. 17:22 ps_ordering

Рисунок Б.29 – Форма сводного финансового отчета по обработке заявок

Приложение В

Фрагменты исходного кода программных модулей АИС «Сибинтек: Управление заявками» на языке программирования высокого уровня

Программный код модуля класса-сущности (на примере правила маршрутизации заявки – RoutingRule.cs)

```
using System;
using System.Windows.Forms;

namespace CRA.Entities
{
    /// <summary>
    /// Класс правила маршрутизации заявки
    /// </summary>
    public class RoutingRule : Interfaces.IDBManagement, Interfaces.IUserManagement
    {
        /// <summary>
        /// Конструктор
        /// </summary>
        /// <param name="__id_">Необязательный идентификатор</param>
        public RoutingRule(int __id_ = 0)
        {
            // Инициализация атрибутов
            Initialize(__id_);
        }

        // -----
        // Атрибуты
        // -----
        private int __id;
        private Client __client;
        private Type __type;
        private Priority __priority;
        private Employee __executor;
        private int __registertype;
        private string __name;

        // ----- Доступ к атрибутам -----
        /// <summary>
        /// Доступ к идентификатору
        /// </summary>
        public int ID { get { return __id; } }
        /// <summary>
        /// Доступ к наименованию
        /// </summary>
        public string Name { get { return __name; } }
        /// <summary>
        /// Доступ к типу регистрации заявки
        /// </summary>
        public int RegisterType { get { return __registertype; } }
        /// <summary>
        /// Доступ к типу регистрации заявки
        /// </summary>
        public Client OrderClient { get { return __client; } }
        /// <summary>
        /// Доступ к типу регистрации заявки
    }
}
```

Продолжение Приложения В

```
/// </summary>
public Type OrderType { get { return __type; } }
/// <summary>
/// Доступ к типу регистрации заявки
/// </summary>
public Priority OrderPriority { get { return __priority; } }
/// <summary>
/// Доступ к типу регистрации заявки
/// </summary>
public Employee OrderExecutor { get { return __executor; } }
// -----

// РЕАИЗАЦИЯ ИНТЕРФЕЙСА С БД -----

/// <summary>
/// Адаптер подключения к БД
/// </summary>
ps_orderingDataSetTableAdapters.routeruleTableAdapter ADAPTER =
    new ps_orderingDataSetTableAdapters.routeruleTableAdapter();

/// <summary>
/// Инициализация данных
/// </summary>
/// <param name="__id_">Необязательный идентификатор</param>
/// <returns>Результат операции</returns>
public bool Initialize(int __id_ = 0)
{
    try
    {
        // Инициализация по идентификатору
        if (__id_ != 0)
        {
            // Получение данных из таблицы
            ps_orderingDataSet.routeruleDataTable TABLE =
ADAPTER.GetDataByID(__id_);
            // Чтение полученных данных
            if (TABLE.Rows.Count > 0)
            {
                __id = __id_;
                __name = Convert.ToString(TABLE.Rows[0].ItemArray[1]);
                __type = (TABLE.Rows[0].ItemArray[2] == DBNull.Value)
                    ? null
                    : new Type(Convert.ToInt32(TABLE.Rows[0].ItemArray[2]));
                __registertype = Convert.ToInt32(TABLE.Rows[0].ItemArray[3]);
                __client = (TABLE.Rows[0].ItemArray[4] == DBNull.Value)
                    ? null
                    : new Client(Convert.ToInt32(TABLE.Rows[0].ItemArray[4]));
                __priority = (TABLE.Rows[0].ItemArray[5] == DBNull.Value)
                    ? null
                    : new Priority(Convert.ToInt32(TABLE.Rows[0].ItemArray[5]));
                __executor = new
Employee(Convert.ToInt32(TABLE.Rows[0].ItemArray[6]));
                // Освободить таблицу
                TABLE.Dispose();
            }
            else
            {
                MessageBox.Show("Запись не найдена!", Application.ProductName,
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```

Продолжение Приложения В

```
        // Возврат неудачи
        return false;
    }
}
else // Инициализация по умолчанию
{
    __executor = new Employee();
    __priority = null;
    __type = null;
    __client = null;
    __registertype = -1;
}
// Возврат успеха
return true;
}
catch (Exception ex)
{
    MessageBox.Show("Ошибка при выполнении операции: " + ex.Message,
Application.ProductName, MessageBoxButtons.OK, MessageBoxIcon.Error);
    // Возврат неудачи
    return false;
}
}
// -----
/// <summary>
/// Вставка записи в БД
/// </summary>
/// <returns>Результат операции</returns>
public bool InsertRow()
{
    try
    {
        // Выполнение команды через адаптер таблицы
        ADAPTER.InsertQuery(Name,
            OrderType != null ? OrderType.ID : (int?)null,
            RegisterType,
            OrderClient != null ? OrderClient.ID : (int?)null,
            OrderPriority != null ? OrderPriority.ID : (int?)null,
            OrderExecutor.ID
        );
        // Возврат успеха
        return true;
    }
    catch (Exception ex)
    {
        MessageBox.Show("Ошибка при выполнении операции: " + ex.Message,
Application.ProductName, MessageBoxButtons.OK, MessageBoxIcon.Error);
        // Возврат неудачи
        return false;
    }
}
// -----
/// <summary>
/// Обновление записи в БД
/// </summary>
/// <returns>Результат операции</returns>
public bool UpdateRow()
{
    try
    {
        // Выполнение команды через адаптер таблицы
```

Продолжение Приложения В

```
        ADAPTER.UpdateQuery(Name,
            OrderType != null ? OrderType.ID : (int?)null,
            RegisterType,
            OrderClient != null ? OrderClient.ID : (int?)null,
            OrderPriority != null ? OrderPriority.ID : (int?)null,
            OrderExecutor.ID, ID);
        // Возврат успеха
        return true;
    }
    catch (Exception ex)
    {
        MessageBox.Show("Ошибка при выполнении операции: " + ex.Message,
Application.ProductName,
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        // Возврат неудачи
        return false;
    }
}
// -----
/// <summary>
/// Удаление записи из БД
/// </summary>
/// <returns>Результат операции</returns>
public bool DeleteRow()
{
    try
    {
        // Выполнение команды через адаптер таблицы
        ADAPTER.DeleteQuery(ID);
        // Возврат успеха
        return true;
    }
    catch (Exception ex)
    {
        MessageBox.Show("Ошибка при выполнении операции: " + ex.Message,
Application.ProductName, MessageBoxButtons.OK, MessageBoxIcon.Error);
        // Возврат неудачи
        return false;
    }
}
// -----

// РЕАЛИЗАЦИЯ ИНТЕРФЕЙСА С ПОЛЬЗОВАТЕЛЕМ -----

/// <summary>
/// Диалог ввода данных
/// </summary>
Dialog.dialogRoutingRule Dialog = null;

/// <summary>
/// Команда пользователя - создать новый
/// </summary>
/// <param name="add_id">Дополнительный идентификатор</param>
/// <returns>Результат выполнения команды</returns>
public bool CreateNew(int add_id = 0)
{
    try
    {
        // Создание диалога ввода данных
        Dialog = new Dialog.dialogRoutingRule();
    }
}
```

Продолжение Приложения В

```
// Наполнение полей диалога актуальными данными
Dialog._Name = Name;
Dialog._RegistrationType = RegisterType;
Dialog._ClientID = OrderClient == null ? -1 : OrderClient.ID;
Dialog._TypeID = OrderType == null ? -1 : OrderType.ID;
Dialog._PriorityID = OrderPriority == null ? -1 : OrderPriority.ID;
// Организация диалога с пользователем
if (Dialog.ShowDialog() == DialogResult.OK)
{
    // Применение введенных значений
    __name = Dialog._Name;
    __registertype = Dialog._RegistrationType;
    __executor = new Employee(Dialog._ExecutorID);
    __client = Dialog._ClientID == -1 ? null : new
Client(Dialog._ClientID);
    __type = Dialog._TypeID == -1 ? null : new Type(Dialog._TypeID);
    __priority = Dialog._PriorityID == -1 ? null : new
Priority(Dialog._PriorityID);
    // Выполнить вставку в БД
    InsertRow();
}
else
{
    // Возврат неудачи - отмена действия пользователем
    return false;
}
// Возврат успеха
return true;
}
catch (Exception ex)
{
    MessageBox.Show("Ошибка при выполнении операции: " + ex.Message,
Application.ProductName,
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    // Возврат неудачи
    return false;
}
}
/// <summary>
/// Команда пользователя - редактировать
/// </summary>
/// <param name="add_id">Дополнительный идентификатор</param>
/// <returns>Результат выполнения команды</returns>
public bool EditData(int add_id = 0)
{
    try
    {
        // Создание диалога ввода данных
        Dialog = new Dialog.dialogRoutingRule
(OrderType == null ? 0 : OrderType.ID,
    OrderPriority == null ? -1 : OrderPriority.ID,
    OrderClient == null ? -1 : OrderClient.ID, OrderExecutor.ID);
        // Наполнение полей диалога актуальными данными
        Dialog._Name = Name;
        Dialog._RegistrationType = RegisterType;
        Dialog._ClientID = OrderClient == null ? -1 : OrderClient.ID;
        Dialog._TypeID = OrderType == null ? -1 : OrderType.ID;
        Dialog._PriorityID = OrderPriority == null ? -1 : OrderPriority.ID;
        // Организация диалога с пользователем
        if (Dialog.ShowDialog() == DialogResult.OK)
```

Продолжение Приложения В

```
{
    // Применение введенных значений
    __name = Dialog._Name;
    __registertype = Dialog._RegistrationType;
    __executor = new Employee(Dialog._ExecutorID);
    __client = Dialog._ClientID == -1 ? null : new
Client(Dialog._ClientID);
    __type = Dialog._TypeID == -1 ? null : new Type(Dialog._TypeID);
    __priority = Dialog._PriorityID == -1 ? null : new
Priority(Dialog._PriorityID);
    // Выполнить обновление в БД
    UpdateRow();
}
else
{
    // Возврат неудачи - отмена действия пользователем
    return false;
}
// Возврат успеха
return true;
}
catch (Exception ex)
{
    MessageBox.Show("Ошибка при выполнении операции: " + ex.Message,
Application.ProductName, MessageBoxButtons.OK, MessageBoxIcon.Error);
    // Возврат неудачи
    return false;
}
}
/// <summary>
/// Команда пользователя - удалить
/// </summary>
/// <returns>Результат операции</returns>
public bool Delete()
{
    try
    {
        // Запрос на удаление
        if (MessageBox.Show("Подтверждаете удаление выбранной записи ?",
"Удаление", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            // Удалить из БД
            DeleteRow();
            // Возврат успеха
            return true;
        }
        // Возврат неудачи - отмена действия пользователем
        return false;
    }
    catch (Exception ex)
    {
        MessageBox.Show("Ошибка при выполнении операции: " + ex.Message,
Application.ProductName, MessageBoxButtons.OK, MessageBoxIcon.Error);
        // Возврат неудачи
        return false;
    }
}
// -----

/// <summary>
```

Продолжение Приложения В

```
/// Поиск правила по критериям
/// </summary>
/// <param name="__type_id_">ID типа заявки</param>
/// <param name="__priority_id_">ID приоритета заявки</param>
/// <param name="__client_id_">ID клиента</param>
/// <param name="__reg_type_">Тип регистрации заявки</param>
/// <returns>Первое правило, найденное в наборе правил, соответствующее критериям
поиска</returns>
public static RoutingRule FindRule(int __type_id_, int __priority_id_, int
__client_id_, int __reg_type_)
{
    // Адаптер подключения к БД - таблица аккаунтов
    ps_orderingDataSetTableAdapters.routeruleTableAdapter TADAPTER =
        new ps_orderingDataSetTableAdapters.routeruleTableAdapter();
    // Возвращаемое значение
    RoutingRule rr_res = null;
    // Поиск правила
    try
    {
        // Получение данных по критериям из таблицы правил
        ps_orderingDataSet.routeruleDataTable TABLE =
            TADAPTER.GetRuleByCriterias(__type_id_, __reg_type_, __priority_id_,
__client_id_);
        // Чтение полученных данных
        if (TABLE.Rows.Count > 0)
        {
            // Создаем объект первого найденного правила
            rr_res = new RoutingRule(Convert.ToInt32(TABLE.Rows[0].ItemArray[0]));
            // Освободить таблицу
            TABLE.Dispose();
        }
        else // Возврат неудачи
            return null;
    }
    catch (Exception)
    {
        // Возврат неудачи
        return null;
    }
    // Возврат значения
    return rr_res;
}
}
```

Программный код подсистемы пользователя – руководителя: модулей формирования отчетов (reportFinancial.cs, reportOrderingTrend.cs, reportStatistic.cs)

```
using System;
using System.IO;
using System.Windows.Forms;

namespace CRA.Subsystems.Functions.Reports
{
    /// <summary>
```

Продолжение Приложения В

```
/// Панель подсистемы создания финансового отчета
/// </summary>
public partial class reportFinancial : UserControl
{
    /// <summary>
    /// Конструктор
    /// </summary>
    public reportFinancial()
    {
        /// Создание и инициализация компонентов
        InitializeComponent();
        /// По умолчанию - текущий месяц
        dtpDateFilterFrom.Value = new DateTime(DateTime.Now.Year, DateTime.Now.Month, 1);
        dtpDateFilterTo.Value = new DateTime(DateTime.Now.Year, DateTime.Now.Month,
1).AddMonths(1).AddDays(-1);
        /// Обновление отчета
        UpdateReport();
    }
    /// <summary>
    /// Перевод строки
    /// </summary>
    string NL = Environment.NewLine;
    /// <summary>
    /// Обновление диаграммы
    /// </summary>
    public void UpdateReport()
    {
        /// Установка диапазона дат и параметра фильтрации для выборки
        DateTime dtFrom = chkTodayOnlyFilter.Checked ? DateTime.Now.Date :
dtpDateFilterFrom.Value.Date;
        DateTime dtTo = chkTodayOnlyFilter.Checked ?
DateTime.Now.Date.AddDays(1).AddSeconds(-1) : dtpDateFilterTo.Value.Date;
        int show_all = chkShowAll.Checked ? 1 : 0;
        /// Загрузка данных
        ps_orderingDataSet.orderssummaryDataTable OSDT =
            orderssummaryTableAdapter.GetDataByFilter(dtFrom, dtTo, show_all);

        /// Счетчик строк
        int i = 1;
        /// Переменная для форматированного HTML-отчета
        string HTML = "";

        /// Статистика
        int completed = 0;
        int summ = 0, csumm;

        /// ГЕНЕРАЦИЯ HTML-КОДА
        //-----
        HTML += "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 Transitional//EN\">" + NL;
        HTML += "<html>" + NL;
        HTML += "    <head>" + NL;
        HTML += "        <title></title>" + NL;
        HTML += "    </head>" + NL;
        HTML += "    <body>" + NL;
        HTML += "        <H2 align=center>Отчет по входящим заявкам за период с
<u><i>" +
            dtFrom.ToLongDateString() + "</i></u> по <u><i>" +
            dtTo.ToLongDateString() + "</i></u></H2>" + NL;
    }
}
```

Продолжение Приложения В

```

HTML += "          <table align=\"center\" border=\"1\" cellpadding=\"0\"
cellspacing=\"0\" frame=\"border\" + NL;
HTML += "          rules=\"all\" width=\"100%\"> + NL;
HTML += "          <tr> + NL;
HTML += "              <td align=\"center\" width=60><b>№п/н</b></td> + NL;
HTML += "              <td align=\"center\" width=75><b>№ заявки</b></td> + NL;
HTML += "              <td align=\"center\"><b>Тип заявки</b></td> + NL;
HTML += "              <td align=\"center\"
width=110><b>Поступила</b></td> + NL;
HTML += "              <td align=\"center\"
width=110><b>Запланировано</b></td> + NL;
HTML += "              <td align=\"center\" width=110><b>Решено</b></td> + NL;
HTML += "              <td align=\"center\" width=110><b>Суммарная
стоимость обработки</b></td> + NL;
HTML += "          </tr> + NL;
HTML += "          <tr> + NL;
HTML += "              <td align=\"center\"><b>1</b></td> + NL;
HTML += "              <td align=\"center\"><b>2</b></td> + NL;
HTML += "              <td align=\"center\"><b>3</b></td> + NL;
HTML += "              <td align=\"center\"><b>4</b></td> + NL;
HTML += "              <td align=\"center\"><b>5</b></td> + NL;
HTML += "              <td align=\"center\"><b>6</b></td> + NL;
HTML += "              <td align=\"center\"><b>7</b></td> + NL;
HTML += "          </tr> + NL;

foreach (System.Data.DataRow dr in OSDT.Rows)
{
    HTML += "          <tr> + NL;
    HTML += "          <td align=\"center\"> + i++.ToString() +
"</td> + NL;
    HTML += "          <td> +
Convert.ToInt32(dr.ItemArray[0]).ToString("0000") + "</td> + NL;
    HTML += "          <td> + Convert.ToString(dr.ItemArray[5]) +
"</td> + NL;
    HTML += "          <td> +
Convert.ToDateTime(dr.ItemArray[1]).ToLongDateString() + "</td> + NL;
    HTML += "          <td> +
Convert.ToDateTime(dr.ItemArray[2]).ToLongDateString() + "</td> + NL;
    // Дата решения заявки
    if (Convert.ToString(dr.ItemArray[3]) != string.Empty)
    {
        HTML += "          <td> +
Convert.ToDateTime(dr.ItemArray[3]).ToLongDateString() + "</td> + NL;
        completed++;
    }
    else
        HTML += "          <td> - </td> + NL;
    // Стоимость заявки
    Int32.TryParse(Convert.ToString(dr.ItemArray[4]), out csumm);
    summ += csumm;
    HTML += "          <td> + csumm.ToString("C0") + "</td> + NL;
    HTML += "          </tr> + NL;
}

HTML += "          </table> + NL;
HTML += "          <br /><br /> + NL;
HTML += "$"          <u><em>Всего заявок обработано за указанный
период:</em></u> <b>{i - 1}</b><br /> + NL;
HTML += "$"          <u><em>из них решено:</em></u> <b>{completed}</b><br /> + NL;
HTML += "$"          <u><em>Общая сумма по обработанным заявкам за указанный
период:</em></u> <b>{summ}</b><br /> + NL;

```

Продолжение Приложения В

```
HTML += "          <br /><br />" + NL;

HTML += "          <div align=\"right\"><em>Руководитель отдела обработки
заявок </em> /_____</br />" + NL;
HTML += "          <br /><em>" + DateTime.Now.Date.ToString("D") +
"</em></div>" + NL;
HTML += "          </body>" + NL;
HTML += "</html>" + NL;

// Освобождение таблиц
OSDT.Dispose();

// Отображение отчета
webbDocument.DocumentText = HTML;
}

/// <summary>
/// Обновление отчета
/// </summary>
/// <param name="sender">Объект события</param>
/// <param name="e">Параметры события</param>
private void lnkFilterOK_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    // Обновление отчета
    UpdateReport();
}

/// <summary>
/// Сохранение содержимого документа в HTML-файл
/// </summary>
/// <param name="sender">Объект, вызвавший событие</param>
/// <param name="e">Параметры события</param>
private void lnkSave_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    SaveFileDialog SD = new SaveFileDialog();
    SD.InitialDirectory = Application.StartupPath;
    SD.Filter = "HyperText Markup Language files (*.html)|*.html";
    if (SD.ShowDialog() == DialogResult.OK)
    {
        String FileName = SD.FileName;
        FileStream fs = new FileStream(FileName, FileMode.OpenOrCreate);
        StreamWriter sw = new StreamWriter(fs);
        sw.WriteLine(this.webbDocument.DocumentText);
        sw.Close();
        fs.Close();
    }
}

/// <summary>
/// Вызов окна предпросмотра документа перед печатью
/// </summary>
/// <param name="sender">Объект, вызвавший событие</param>
/// <param name="e">Параметры события</param>
private void lnkPreview_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    webbDocument.ShowPrintPreviewDialog();
}

/// <summary>
/// Вызов диалога печати документа
/// </summary>
/// <param name="sender">Объект, вызвавший событие</param>
/// <param name="e">Параметры события</param>
```

Продолжение Приложения В

```
private void lnkPrint_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    webbDocument.ShowPrintDialog();
}
}

using System;
using System.Windows.Forms;

namespace CRA.Subsystems.Functions.Reports
{
    /// <summary>
    /// Панель подсистемы создания отчета по статистике решения заявок
    /// </summary>
    public partial class reportStatistic : UserControl
    {
        /// <summary>
        /// Конструктор
        /// </summary>
        public reportStatistic()
        {
            // Создание и инициализация компонентов
            InitializeComponent();
            // Обновление данных
            UpdateChartCountByMonths();
            UpdateChartSummaryByMonths();
            UpdateChartGroupingByType();
            UpdateChartGroupingByStatus();
            UpdateChartGroupingByExecutor();
        }

        // Названия месяцев
        string[] months = { "Янв.", "Фев.", "Мар.", "Апр.", "Май", "Июнь", "Июль",
            "Авг.", "Сен.", "Окт.", "Ноя.", "Дек." };

        /// <summary>
        /// Обновление диаграммы "по исполнителям"
        /// </summary>
        public void UpdateChartGroupingByExecutor()
        {
            reportgroupordersbyexecutorTableAdapter.
            Fill(ps_orderingDataSet.reportgroupordersbyexecutor);
        }
        /// <summary>
        /// Обновление диаграммы "по статусу"
        /// </summary>
        public void UpdateChartGroupingByStatus()
        {
            reportgroupordersbystatusTableAdapter.
            Fill(ps_orderingDataSet.reportgroupordersbystatus);
        }
        /// <summary>
        /// Обновление диаграммы "по типу"
        /// </summary>
        public void UpdateChartGroupingByType()
        {
            reportgroupordersbytypeTableAdapter.
            Fill(ps_orderingDataSet.reportgroupordersbytype);
        }
    }
}
```

Продолжение Приложения В

```
/// <summary>
/// Обновление диаграммы "суммы по месяцам"
/// </summary>
public void UpdateChartSummaryByMonths()
{
    // По месяцам
    for (int m = 1, v = 0; m <= 12; m++)
    {
        // Получение значения
        ps_orderingDataSet.reportorderssummaryinmonthsDataTable dt =
            reportorderssummaryinmonthsTableAdapter.GetDataByMonth(m,
DateTime.Now.Year);
        // Чтение значения
        if (dt.Rows.Count < 1)
            v = 0;
        else
            v = Convert.ToInt32(dt.Rows[0].ItemArray[0]);
        // Добавление в чарт
        chartSumOfMonths.Series["Series1"].Points.AddXY(months[m - 1], v);
    }
}
/// <summary>
/// Обновление диаграммы "количество по месяцам"
/// </summary>
public void UpdateChartCountByMonths()
{
    // По месяцам
    for (int m = 1, v = 0; m <= 12; m++)
    {
        // Получение значения
        ps_orderingDataSet.reportorderscountinmonthDataTable dt =
            reportorderscountinmonthTableAdapter.GetDataByMonth(m,
DateTime.Now.Year);
        // Чтение значения
        if (dt.Rows.Count < 1) v = 0;
        else v = Convert.ToInt32(dt.Rows[0].ItemArray[0]);
        // Добавление в чарт
        chartCountOfMonths.Series["Series1"].Points.AddXY(months[m - 1], v);
    }
}
}
}
using System;
using System.Windows.Forms;

namespace CRA.Subsystems.Functions.Reports
{
    /// <summary>
    /// Панель подсистемы создания отчета по динамике решения заявок
    /// </summary>
    public partial class reportOrderingTrend : UserControl
    {
        /// <summary>
        /// Конструктор
        /// </summary>
        public reportOrderingTrend()
        {
            // Создание и инициализация компонентов
            InitializeComponent();
            // По умолчанию - текущий месяц
        }
    }
}
```

Продолжение Приложения В

```
        dtpDateFilterFrom.Value = new DateTime(DateTime.Now.Year, DateTime.Now.Month, 1);
        dtpDateFilterTo.Value = new DateTime(DateTime.Now.Year, DateTime.Now.Month,
1).AddMonths(1).AddDays(-1);
        // Обновление диаграммы
        UpdateChart();
    }

    /// <summary>
    /// Обновление диаграммы
    /// </summary>
    public void UpdateChart()
    {
        // Установка диапазона дат для выборки
        DateTime dtFrom = chkTodayOnlyFilter.Checked ? DateTime.Now.Date :
dtpDateFilterFrom.Value.Date;
        DateTime dtTo = chkTodayOnlyFilter.Checked ?
DateTime.Now.Date.AddDays(1).AddSeconds(-1) : dtpDateFilterTo.Value.Date;
        // Предварительная очистка тренда
        chartOrdersDynamic.Series["SerieNew"].Points.Clear();
        chartOrdersDynamic.Series["SerieCompleted"].Points.Clear();
        // Наполнение данными
        long no, co;
        DateTime dt = dtFrom;
        while (dt <= dtTo)
        {
            // Чтение данных за выбранную дату
            no = orderTableAdapter.GetCountOfNewOrders(dt.Day, dt.Month, dt.Year) ?? 0;
            co = orderTableAdapter.GetCountOfCompletedOrders(dt.Day, dt.Month, dt.Year) ?? 0;
            // и добавление их в соответствующий тренд
            chartOrdersDynamic.Series["SerieNew"].Points.AddXY(dt, no);
            chartOrdersDynamic.Series["SerieCompleted"].Points.AddXY(dt, co);
            // Следующая дата
            dt = dt.AddDays(1);
        }
    }
    /// <summary>
    /// Обновление диаграммы
    /// </summary>
    /// <param name="sender">Объект события</param>
    /// <param name="e">Параметры события</param>
    private void lnkFilterOK_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
    {
        // Обновление диаграммы
        UpdateChart();
    }
}
}
```