

Министерство образования и науки Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

**ИНСТИТУТ ФИЗИКИ, МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

Прикладная математика и информатика

09.03.03 Прикладная информатика

Бизнес-информатика

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему **Разработка ИС информационно-технической поддержки
пользователей для ИТ-компании на примере ЧП «Net Soft System»**

Студент Александр Владимирович Нетаев _____

(личная подпись)

Руководитель Сергей Вазгенович Мкртычев _____

(личная подпись)

Консультанты _____

(И.О. Фамилия)

(личная подпись)

(И.О. Фамилия)

(личная подпись)

Допустить к защите

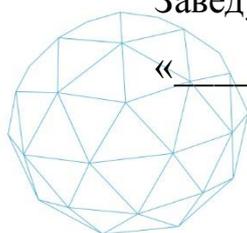
Заведующий кафедрой к.т.н., доцент, А. В. Очеповский _____

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

(личная подпись)

« _____ » _____ 2018г.

Тольятти 2018



АННОТАЦИЯ

Выпускная бакалаврская работа выполнена на тему «Разработка ИС информационно-технической поддержки пользователей для ИТ-компании на примере ЧП «Net Soft System».

Объем представленной работы составляет 60 страниц, включает 26 рисунков, 23 таблицы и 20 источников и литературы.

Целью бакалаврской работы является разработка информационной системы информационно-технической поддержки пользователей ИТ-компании ЧП «Net Soft System».

В работе проводится анализ объекта исследования и бизнес-процессов для выявления недостатков существующего учета, устранить которые необходимо путем внедрения информационной системы.

Разрабатываемая информационная система будет способствовать решению следующих задач: учет клиентов и пользователей; создание заявок и отправка их на регистрацию; назначение/изменение исполнителя заявки; регистрация заявок с присвоением номера; обработка и исполнение заявок; назначение сроков исполнения заявок согласно приоритетам; настройка и продление сроков исполнения заявок; обмен сообщениями; обмен графическими материалами; автоматическая смена статуса заявки; уведомление о новых сообщениях и заявках; формирование расписания задач и контроль их исполнения; учет сотрудников; формирование отчетов и т.д.

Ключевые слова: заявка, техническое обслуживание, информационная система, программное обеспечение, автоматизация, база данных; «Отдел техподдержки».

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
Глава 1. Анализ предметной области.....	6
1.1. Техничко-экономическая характеристика предметной области	6
1.1.1. Характеристика предприятия	6
1.1.2. Краткая характеристика подразделения и его видов деятельности	6
1.1.3. Сущность задачи автоматизации.....	9
1.2. Концептуальное моделирование предметной области.....	10
1.3. Постановка задачи.....	19
1.3.1. Цель и назначение автоматизированного варианта решения задачи..	19
1.3.2. Общая характеристика организации решения задачи на ЭВМ.....	19
1.4. Анализ существующих разработок и обоснование выбора технологии проектирования.....	22
Выводы по главе 1.....	25
Глава 2. Разработка и реализация проектных решений	26
2.1. Логическое моделирование предметной области.....	26
2.1.1. Логическая модель и ее описание.....	26
2.1.2. Используемые классификаторы и системы кодирования.....	32
2.1.3. Характеристика нормативно-справочной и входной оперативной информации	33
2.1.4. Физическая модель базы данных.....	34
2.1.5. Характеристика результатной информации	37
2.2. Физическое моделирование информационной системы	38

2.2.1. Выбор архитектуры информационной системы	38
2.2.2. Функциональная схема проекта.....	44
2.2.3. Описание алгоритма идентификации пользователей	45
2.2.4. Схема взаимосвязи программных модулей и информационных файлов	46
2.3. Технологическое обеспечение задачи	49
2.3.1. Организация технологии сбора, передачи, обработки и выдачи информации	49
2.3.2. Схема технологического процесса сбора, передачи, обработки и выдачи информации.....	49
Выводы по главе 2.....	54
Глава 3. Оценка и обоснование экономической эффективности проекта....	55
3.1. Планирование и контроль выполнения работ	55
3.2 Планируемая себестоимость разработки информационной системы.....	55
3.3. Обоснование экономической целесообразности разработки и внедрения информационной системы	59
Выводы по главе 3.....	60
ЗАКЛЮЧЕНИЕ	61
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	63
ПРИЛОЖЕНИЯ	65
Описание обозначений базы данных	65
Фрагменты листинга программы	66

ВВЕДЕНИЕ

Актуальность темы исследования обусловлена тем, что в век информационных технологий успешное и прибыльное функционирование любого бизнеса невозможно без использования автоматизированных информационных систем.

ИТ-компания ЧП «Net Soft System», деятельность которой состоит в предоставлении ИТ-услуг клиентам, нуждается в автоматизации, т.к. от уровня оказываемых клиенту услуг зависят экономические результаты всей компании.

Объектом исследования бакалаврской работы является бизнес-процесс приема и анализа заявок техподдержки ИТ-компания ЧП «Net Soft System».

Предметом исследования бакалаврской работы является информационная система информационно-технической поддержки пользователей ИТ-компания ЧП «Net Soft System».

Целью бакалаврской работы является разработка информационной системы информационно-технической поддержки пользователей ИТ-компания ЧП «Net Soft System».

Для достижения поставленных задач в работе, необходимо решить следующие задачи:

- провести анализ предметной области с целью выявления процессов учета, требующих автоматизации;
- описать функциональные требования к проектируемой системе;
- провести анализ средств разработки приложения и базы данных;
- построить модель данных на логическом и физическом уровнях;
- разработать информационную систему (ИС) и описать принципы работы с ней;
- оценить себестоимость информационной системы и целесообразность ее разработки.

Структура дипломной работы состоит из введения, трех глав, заключения, списка использованной литературы и литературы.

В первой главе дано описание деятельности ЧП «Net Soft System», представлена его оргструктура. Описан существующий бизнес-процесс обработки заявок и выявлены его недостатки. Разработана модель автоматизированного бизнес-процесса обработки заявок и сформулированы требования к проектируемой ИС по видам обеспечения.

Во второй главе разработаны логическая и физическая модель данных исследуемой предметной области. Разработано приложение ИС и описана его функциональность.

Третья глава посвящена оценке себестоимости и экономической целесообразности разработки системы.

Глава 1. Анализ предметной области

1.1. Техничко-экономическая характеристика предметной области

1.1.1. Характеристика предприятия

Рынок ИТ-услуг по разработке и сопровождению программного обеспечения (ПО) представляет собой совокупность экономических отношений, возникших в процессе купли-продажи программного продукта.

Основными субъектами рынка ПО являются разработчики и заказчики (пользователи ПО).

ЧП «Net Soft System» является эксклюзивным представителем AMOS на территории Российской Федерации, входящей в международную группу компаний «SpecTec Group Holdings», офисы которой расположены в более чем 20 странах мира [12].

С конца 90-х годов компания внедряет корпоративные информационные системы на основе ПО AMOS в крупнейших Российских компаниях, работающих в сфере транспорта и нефтегазовой промышленности. Проекты, которые реализует ЧП «Net Soft System», успешно работают на практике и доказывают свою эффективность.

Цель функционирования предприятия ЧП «Net Soft System» состоит в уменьшении издержек и максимизации получаемой прибыли. Прибыль компании во многом зависит от уровня удовлетворенности заказчика качеством приобретенного ПО.

В сферу деятельности ЧП «Net Soft System» входят услуги по созданию, сопровождению и модификации ИС, обеспечивающих комплексную автоматизацию сферы бизнеса заказчика.

1.1.2. Краткая характеристика подразделения и его видов деятельности

Для определения списка автоматизируемых задач, необходимо подробнее ознакомиться с организационной структурой компании и выявить перечень подразделений, участвующих в процессе техподдержки пользователей.

На рисунке 1 представлена организационная структура ЧП «Net Soft System» [13].

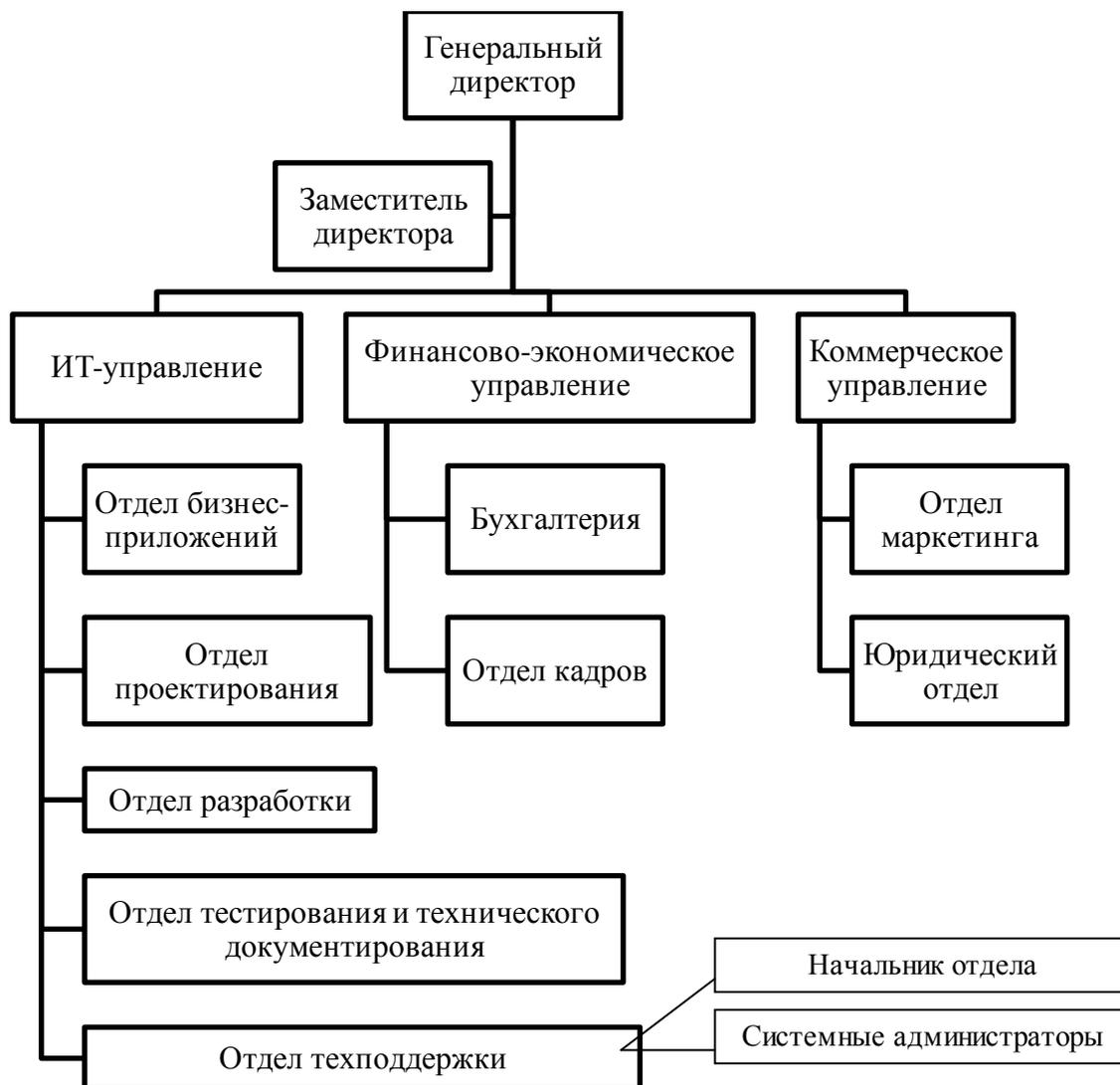


Рисунок 1. Организационная структура ЧП «Net Soft System»

Руководство компанией осуществляет Генеральный директор, который обеспечивает исполнение всех обязательств компании перед государственными бюджетами; решает финансово-экономические и производственно-хозяйственные вопросы предприятия; обеспечивает соблюдение всех, установленных законодательством, условий и требований по охране труда и здоровья сотрудников и т.п. В отсутствие Генерального директора его обязанности исполняет Заместитель Директора.

ИТ-управление является важным звеном в структуре компании, обеспечивающим взаимодействие сотрудников компании с заказчиками.

Основными задачами «ИТ-управления» являются:

- определение требований к ПО,
 - проектирование ПО;
 - разработка ПО и БД; тестирование и документирование ПО;
- техническая поддержка пользователей;

Структура «ИТ-управления» представлена подразделениями, названия которых соответствуют перечню перечисленных задач.

Отдел бизнес-приложений занимается предпроектным обследованием, изучает объект автоматизации и определяет требования к ПО.

Отдел проектирования разрабатывает структуру базы данных, создает макеты программы и т.п.

В отделе разработок работают программисты, которые занимаются непосредственно разработкой программного продукта.

Отдел тестирования и документирования составляет тестовые наборы данных для каждой подсистемы разработанного продукта, осуществляет их проверку и поиск ошибок. Сведения об ошибках и недочетах, выявленные в процессе тестирования, направляются в отдел разработки для их устранения.

После успешного тестирования программного продукта, сотрудники отдела тестирования и документирования разрабатывают инструкции пользователей и готовят проектную документацию.

Отдел техподдержки оказывает техническую поддержку пользователям после внедрения системы. В данном подразделении реализуется бизнес-процесс, который является объектом бакалаврской работы.

1.1.3. Сущность задачи автоматизации

Самый ценный и значимый ресурс любой компании – это клиенты. Стратегия компании по управлению взаимоотношениями с клиентами должна строиться на двух принципах: привлечение новых клиентов и удержание существующих. С точки зрения экономической выгоды, гораздо эффективнее поддерживать сотрудничество с постоянными клиентами, чем искать новых.

Так, Принцип Парето гласит, что около 80% прибыли компании обеспечивается 20% ее клиентов.

Однако, даже работая с существующими клиентами, не надо забывать о конкурентах, которые могут переманить клиента, предлагая ему свой продукт.

Повсеместное распространение Интернет привело к тому, что потенциальный клиент имеет возможность получения доступа к любой части рынка услуг при минимальных затратах.

Сохранение сотрудничества с имеющимися клиентами в условиях жесткой конкуренции рынка является основным приоритетом - важно минимизировать отток потребителей и при этом увеличить их прибыльность.

В наши дни этого не достичь без сохранения доверия и лояльности клиентов. Источником доверия является индивидуальный подход к каждому клиенту и повышение качества информационно-технической поддержки пользователей, что позволило бы заказчику своевременно разрешать проблемы и вопросы, возникающие в процессе эксплуатации программных комплексов и продуктов.

Поэтому для успешного развития ЧП «Net Soft System» в первую очередь необходимо создать условия по эффективному взаимодействию пользователей с отделом техподдержки компании.

1.2. Концептуальное моделирование предметной области

Для описания существующей технологии приема и анализа заявок в отделе техподдержки ЧП «Net Soft System» построим структурно-функциональную диаграмму изучаемого процесса по методологии SADT.

Среди известных CASE-средств, пригодных для реализации таких диаграмм, предпочтение было отдано продукту компании «Computer Associates» - «BPwin». Проектирование функциональных моделей будет производиться на основе стандарта IDEF0.

Одним из способов обращения заказчиков в ЧП «Net Soft System» служит информационный запрос (заявка), где может быть указана информация об ошибках и сбоях в работе системы или же представлены вопросы по работе с программой.

Контекстная диаграмма бизнес-процесса (БП) «Прием и анализ заявок отделом техподдержки ЧП «Net Soft System» AS-IS («КАК ЕСТЬ») представлена на рисунке 2.

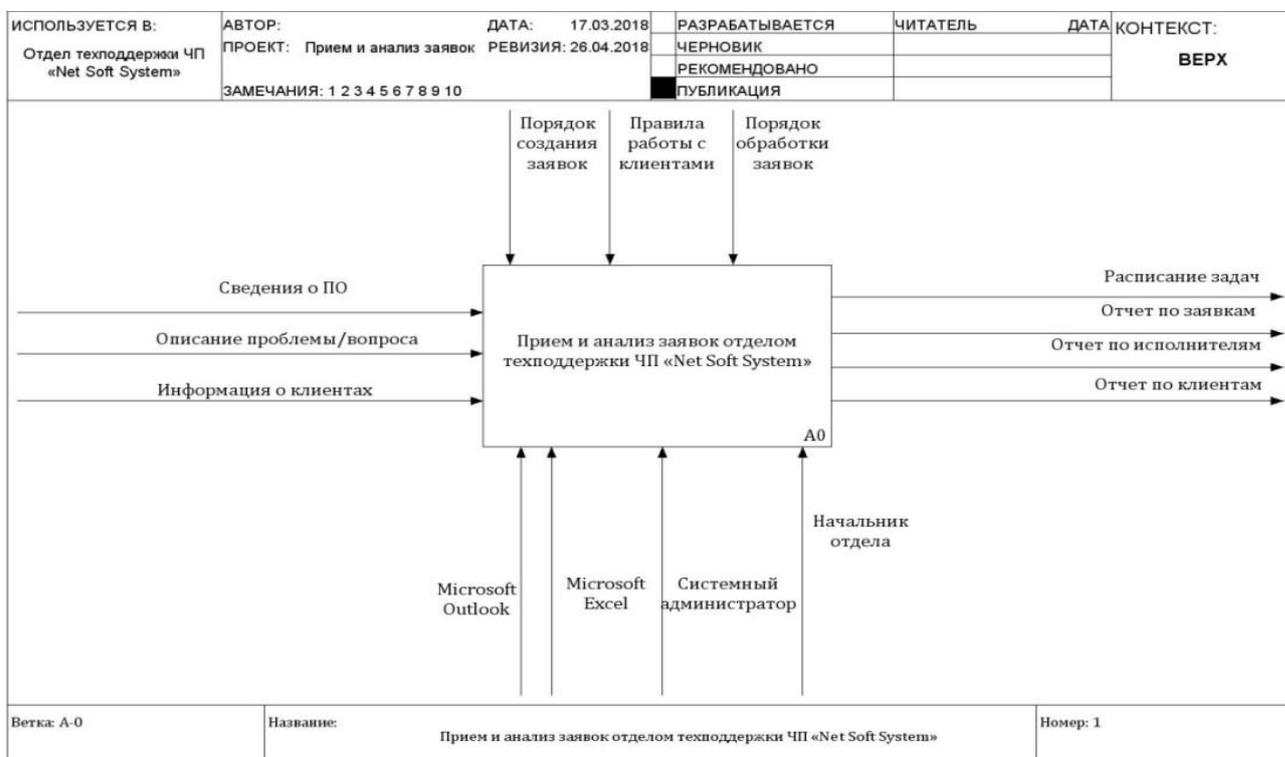


Рисунок 2. Контекстная диаграмма БП «Прием и анализ заявок отделом техподдержки ЧП «Net Soft System». Модель AS-IS

Декомпозиция БП «Прием и анализ заявок отделом техподдержки ЧП «Net Soft System» представлена на рисунке 3.

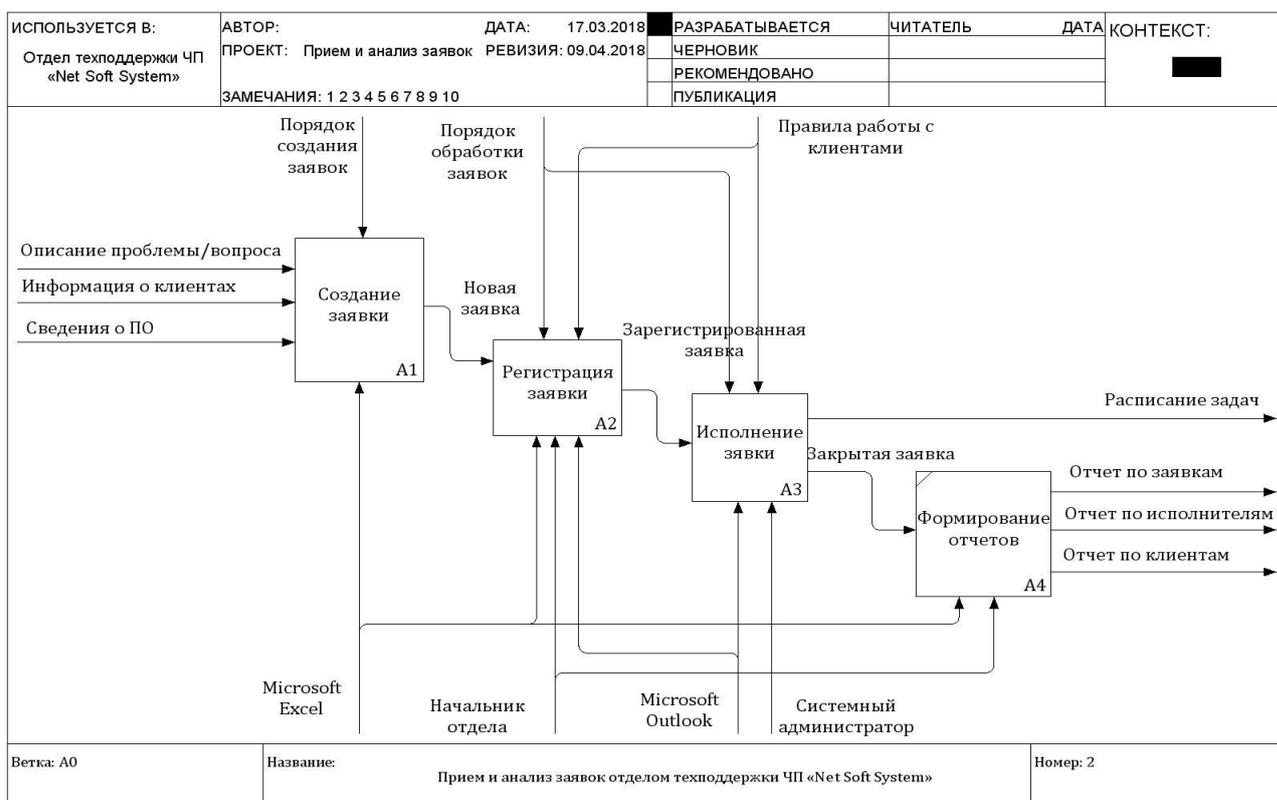


Рисунок 3. Декомпозиция БП «Прием и анализ заявок отделом» техподдержки ЧП «Net Soft System». Модель AS-IS

БП приема и анализа заявок можно разбить на четыре блока, взаимосвязанных друг с другом:

- создание заявки (пользователь заполняет необходимые поля в файле «Макет заявки.xlsx» и отправляет его посредством «Microsoft Outlook» на электронный адрес «Отдела техподдержки» ЧП «Net Soft System»);
- регистрация заявки (отправленный пользователем файл поступает на электронный адрес начальника «Отдела техподдержки», который должен переслать полученную заявку одному из системных администраторов для дальнейшего исполнения);
- исполнение заявки (специалист, назначенный исполнителем заявки, должен приступить к решению вопроса, указанного в заявке);
- формирование отчетов.

Рассмотрим каждый блок в отдельности. Для этого представим декомпозицию БП «Создание заявки» (рисунок 4).

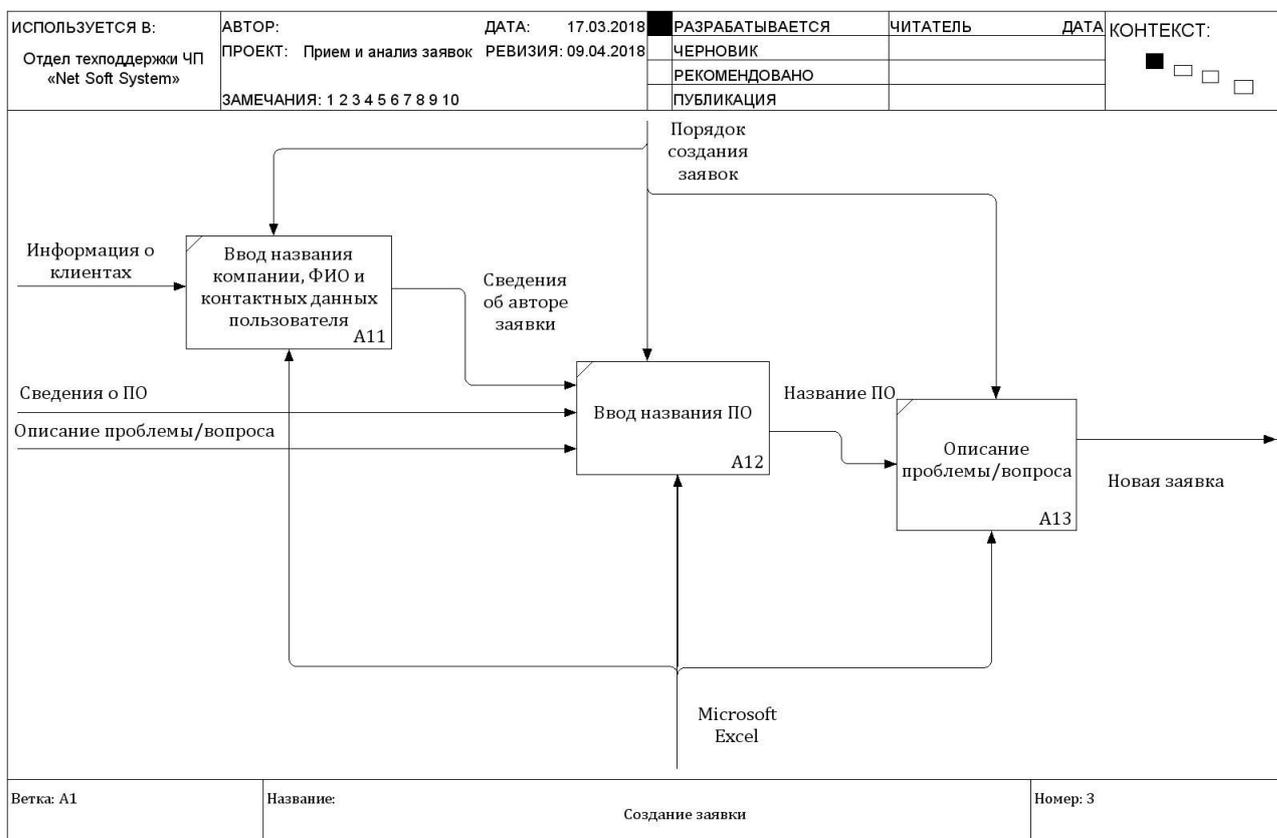


Рисунок 4. Декомпозиция БП «Создание заявки». Модель AS-IS

Все операции в рамках БП «Создание заявки» осуществляются пользователем (клиентом) в Microsoft Excel в файле «Макет заявки.xlsx». Данная операция включает следующие шаги:

- ввод названия компании, ФИО пользователя и его контактных данных (номер телефона и электронный адрес в Microsoft Outlook);
- ввод названия ПО, при работе с которым возникла проблема;
- описание проблемы или вопроса – предполагает подробное изложение сути возникшей проблемы.

Перейдем к блоку «Регистрация заявки», декомпозиция которого представлена на рисунке 5.

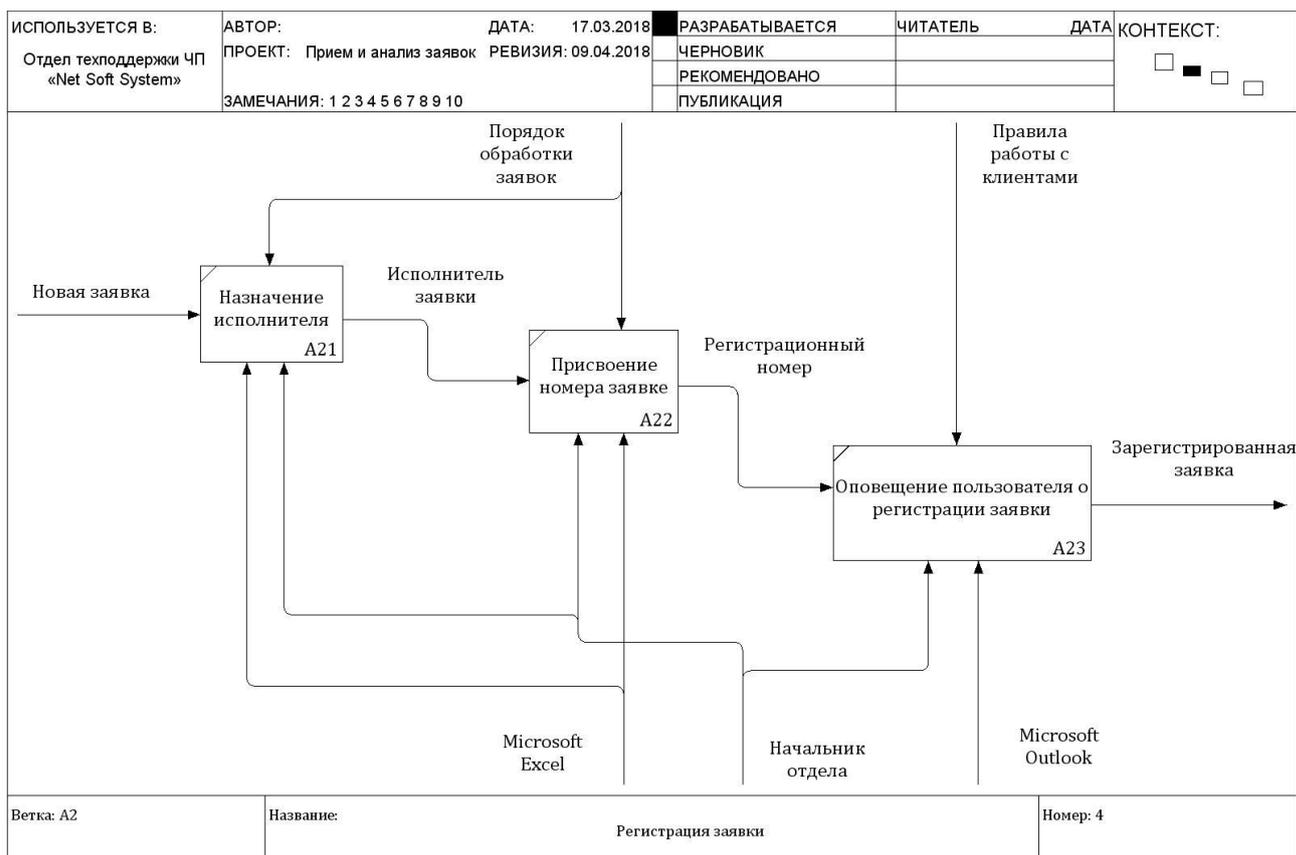


Рисунок 5. Декомпозиция БП «Регистрация заявки». Модель AS-IS

Процедуру по регистрации заявки осуществляет начальник «Отдела техподдержки». Данная операция, согласно существующей технологии, включает следующие шаги:

- назначение исполнителя (при выборе исполнителя необходимо учитывать сколько заявок находится в работе у каждого системного администратора);
- присвоение номера заявке (номер назначается в соответствии с порядковым номером, следующим за предыдущим номером зарегистрированной заявки);
- оповещение пользователя о регистрации заявки.

На рисунке 6 представлена декомпозиция БП «Исполнение заявки».

В операции «Исполнение заявки» участвуют системные администраторы, выполняющие следующие шаги для исполнения заявки:

- оповещение пользователя о начале работы с заявкой;
- обмен сообщениями и графическими материалами (обмен любой информацией должен производиться в «Microsoft Outlook»);
- создание задач (системный администратор должен обговорить с клиентом дату и время продолжения работы по заявке. Выбранное время системный администратор фиксирует в «Microsoft Outlook» в виде задачи);
- исполнение задач;
- оповещение пользователя о завершении работы по заявке.

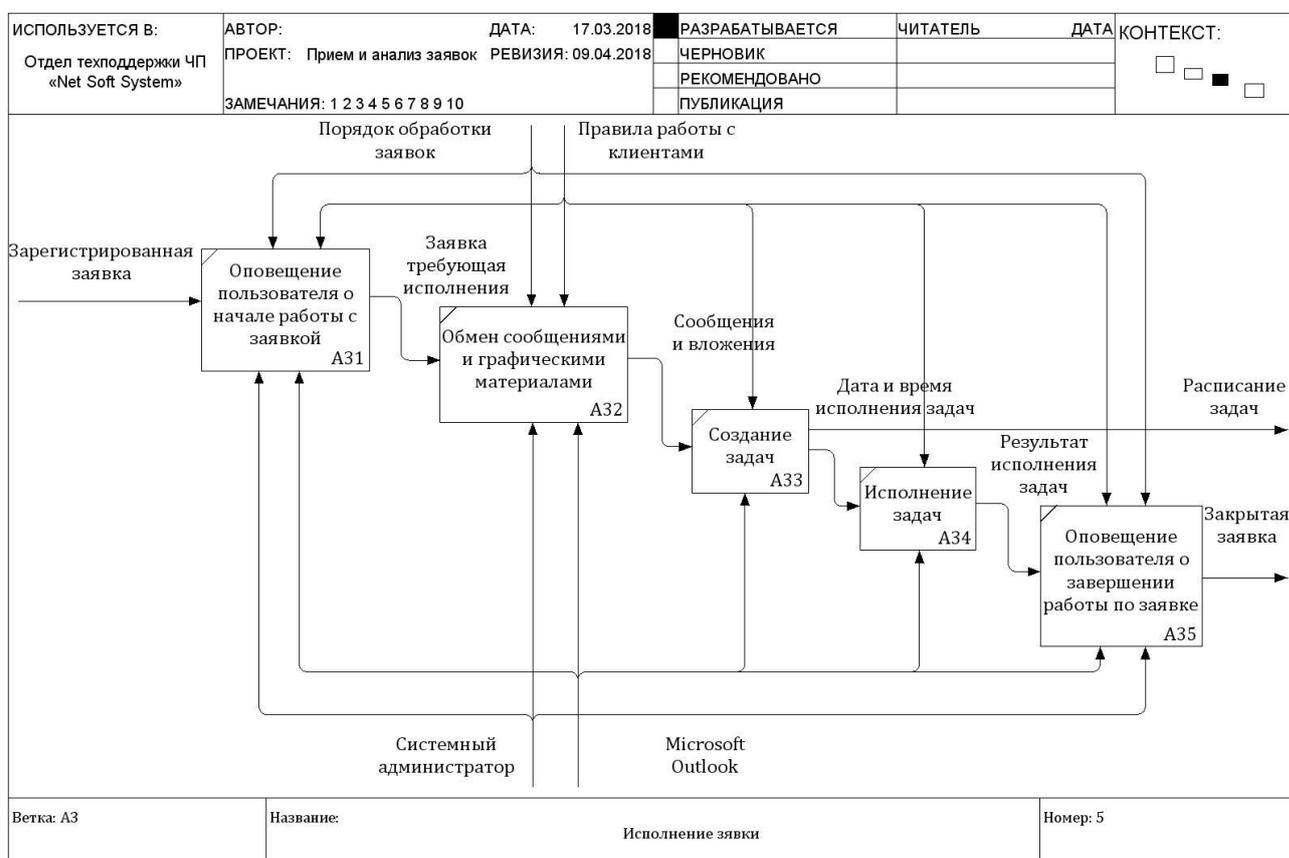


Рисунок 6. Декомпозиция БП «Исполнение заявки». Модель AS-IS

Согласно построенной модели, существующая технология учета заявок отделом техподдержки имеет следующие недостатки:

- не ведется учет данных о клиенте;
- заявки создаются в «Microsoft Excel»;

- обмен сообщениями и графическими материалами осуществляется в «Microsoft Outlook» без возможности привязки сообщений к заявке;
- отсутствует система статусов для заявок, вследствие чего невозможно понять на каком этапе находится исполнение заявки;
- отсутствует система приоритетов для заявок (например, обычный/низкий/высокий);
- не осуществляется назначение сроков исполнения заявок и контроль соблюдения этих сроков;
- отсутствует перечень тем для заявок, это приводит к некорректной и неоднозначной формулировке заголовков заявок;
- отсутствуют сведения, позволяющие грамотно распределять заявки между исполнителями, так как не ведется журнал занятости сотрудников;
- нет уведомлений о новых заявках;
- сотрудникам приходится вручную отправлять пользователям письма, содержащие информацию о ходе обработки заявки;
- высокая трудоемкость и низкая оперативность обработки информации;
- недостоверные результаты, представленные в отчетах.

В связи с этим, компании необходима информационная система, способная устранить существующие проблемы учета, повысить качество и скорость обработки заявок, поступающих от клиентов.

Предлагаемая технология выполнения бизнес-процесса «Прием и анализ заявок отделом техподдержки ЧП «Net Soft System» представляет собой описание процесса учета заявок после внедрения проектируемой информационной системы.

Декомпозиция БП «Прием и анализ заявок отделом техподдержки ЧП «Net Soft System» ТО-ВЕ («КАК ДОЛЖНО БЫТЬ») согласно предлагаемой технологии представлена на рисунке 7.

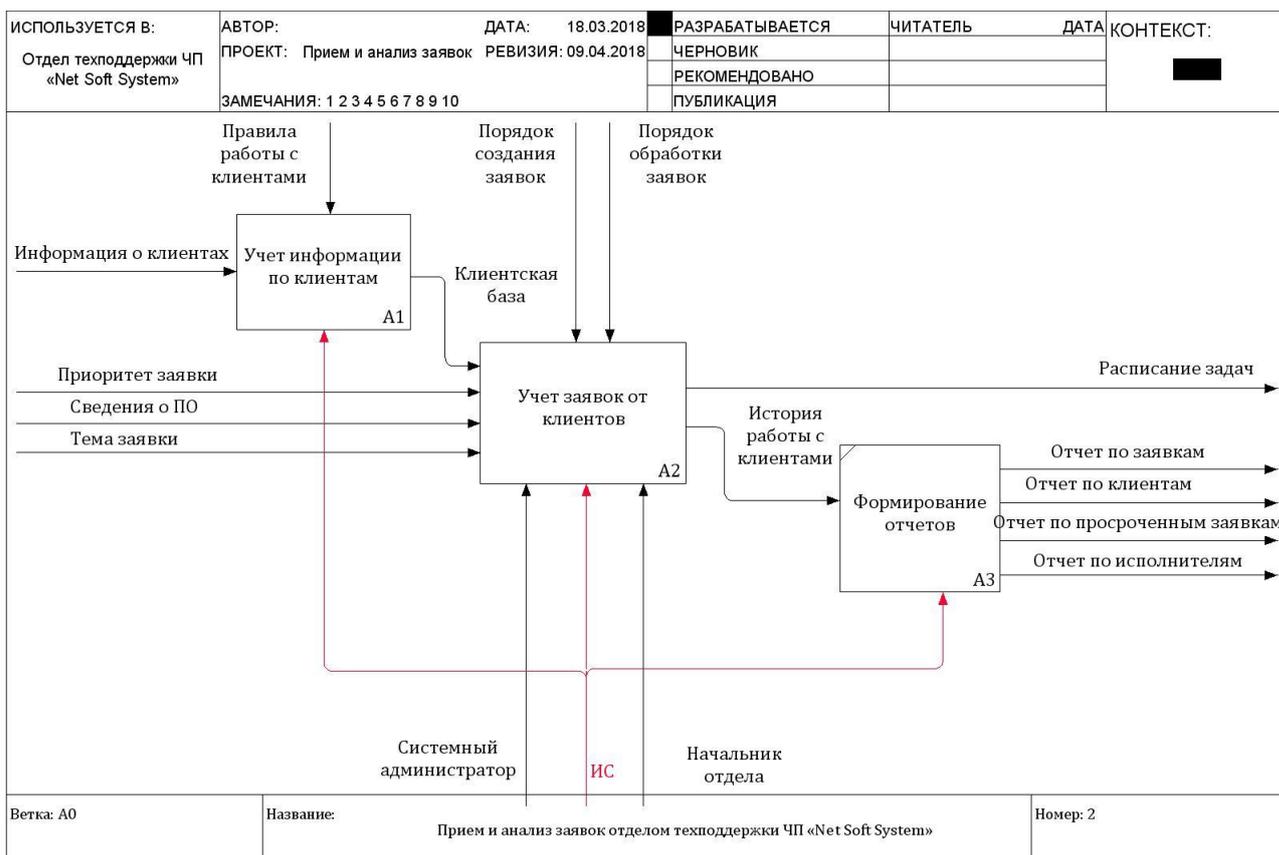


Рисунок 7. Декомпозиция БП «Прием и анализ заявок отделом техподдержки ЧП «Net Soft System». Модель TO-BE

Прием и анализ заявок, согласно новой технологии, должен начинаться со сбора сведений о клиентах и дальнейшего использования полученных данных при взаимодействии с клиентами (блок «Учет заявок от клиентов») и анализе деятельности компании на основе данных об истории работы с клиентами (блок «Формирование отчетов»).

На рисунке 8 изображена схема потока данных операции «Создание заявки» согласно предлагаемой технологии.

Данная операция включает следующие шаги:

- определение параметров пользователя (система автоматизировано производит идентификацию пользователя);
- выбор приоритета, названия ПО и темы (выбор из списка);
- заполнение краткого и полного содержания заявки (клавиатурный ввод).

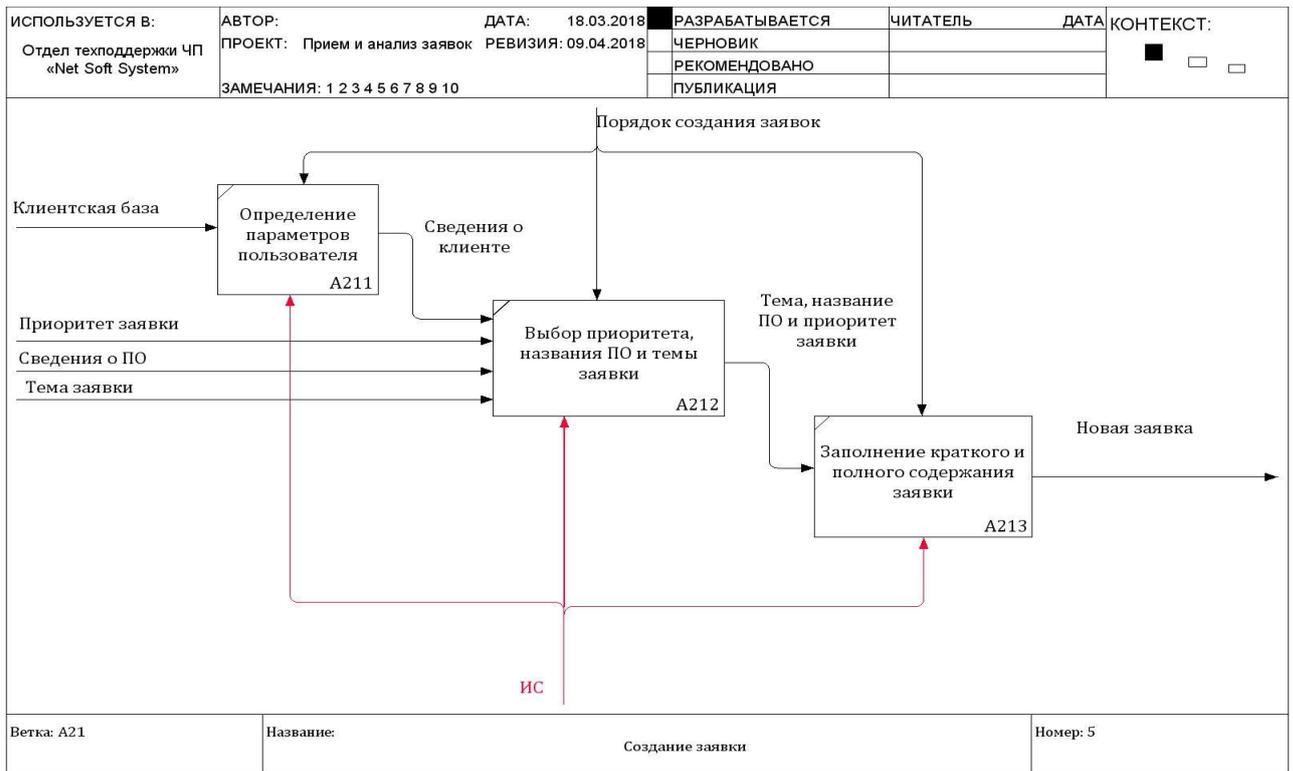


Рисунок 8. Декомпозиция БП «Создание заявки». Модель ТО-ВЕ
 На рисунке 9 изображена декомпозиция БП «Регистрация заявки».

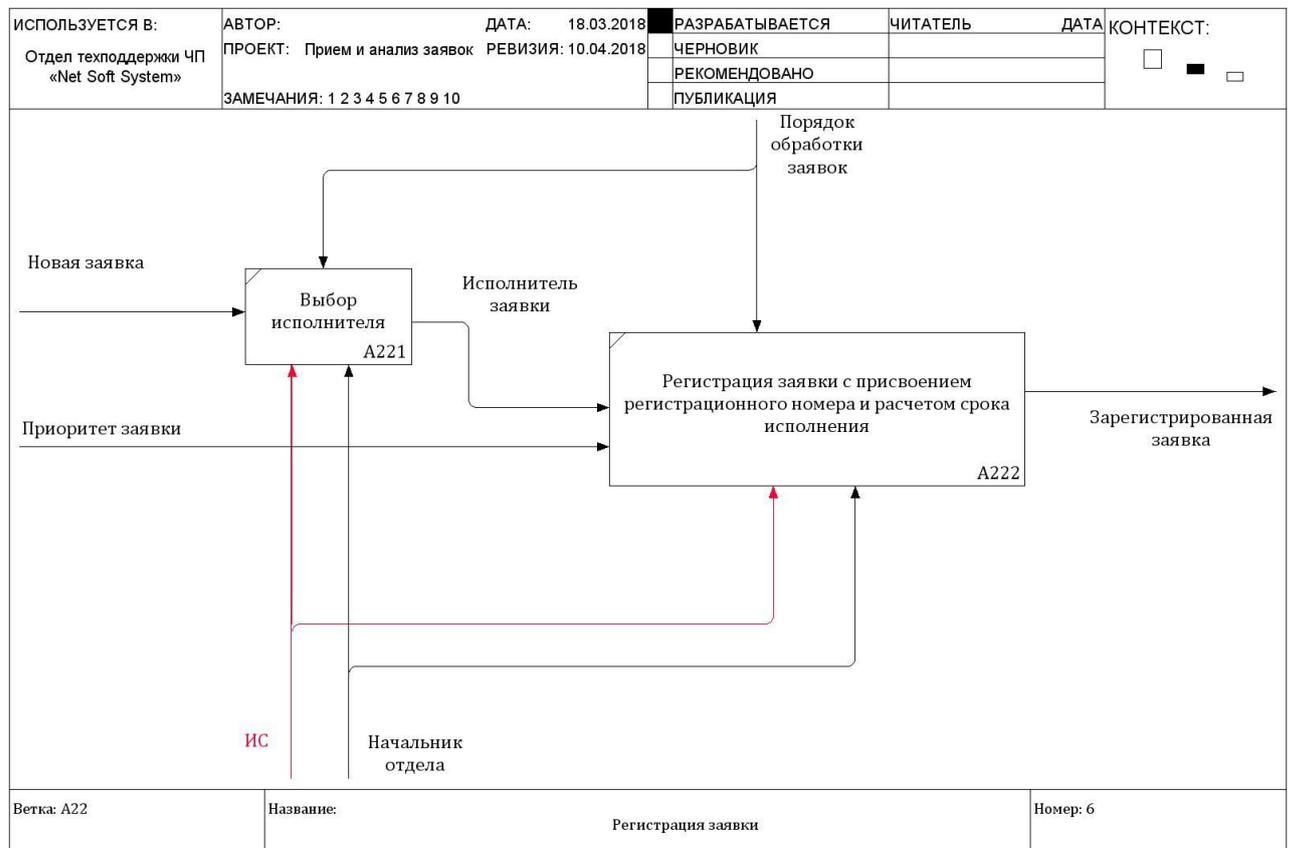


Рисунок 9. Декомпозиция БП «Регистрация заявки». Модель ТО-ВЕ

Новая технология регистрации заявок намного упростилась по сравнению с существующей, за счет:

- выбора исполнителя на основе автоматизированного формирования отчетности о занятости исполнителей заявок;
- автоматизированного присвоения регистрационного номера заявке (в соответствии с порядковым номером предыдущей заявки);
- автоматизированного расчёта срока исполнения заявки (в соответствии с приоритетом заявки).

Декомпозиция БП «Исполнение заявки», по предлагаемой технологии, представлена на рисунке 10.

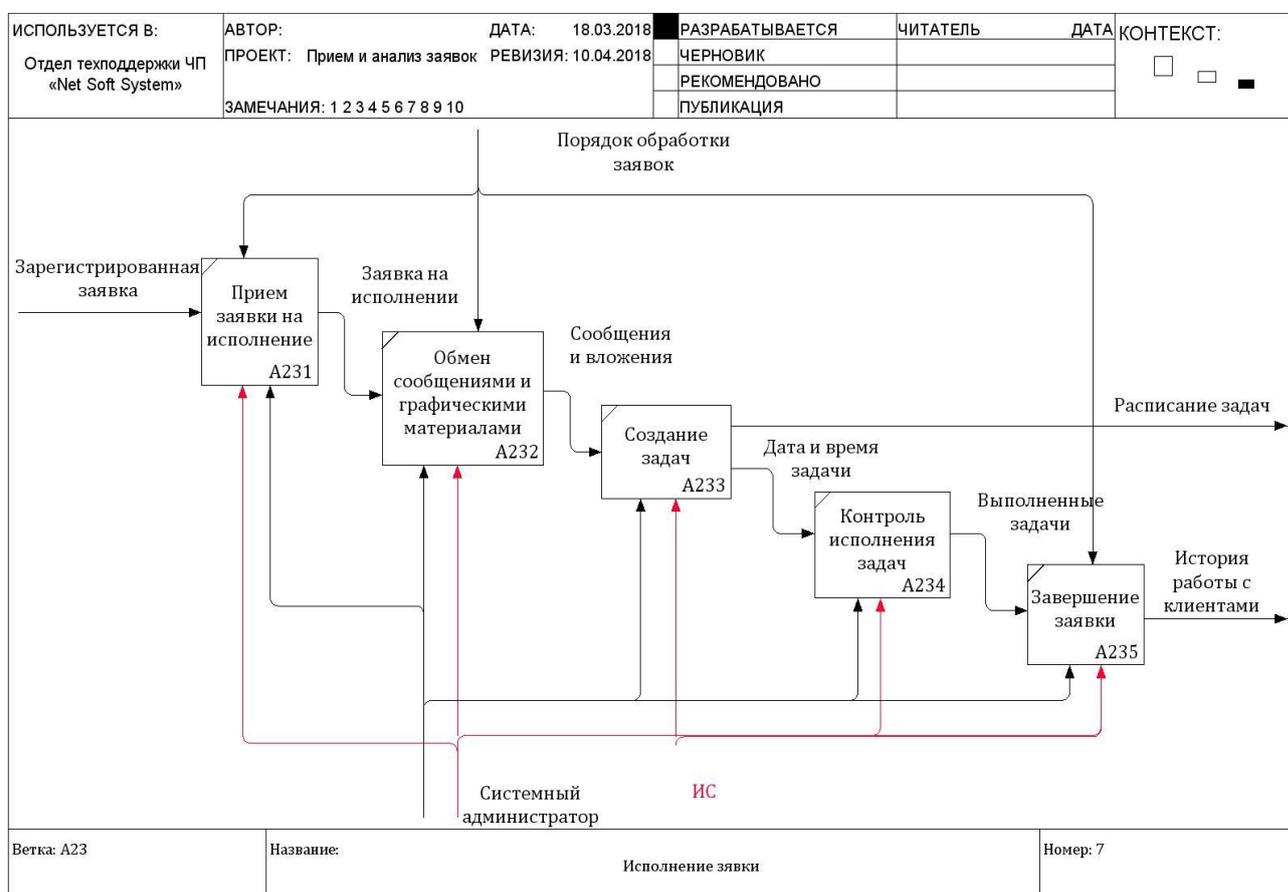


Рисунок 10. Декомпозиция процесса «Исполнение заявки». Модель ТО-ВЕ

Согласно предлагаемой технологии, обмен сообщениями и графическими материалами (например, скриншотами ошибок) должен производиться только в ИС.

Также в системе будет доступна возможность создания задач и контроля их исполнения, что избавляет системного администратора от необходимости обращения к, используемой ранее, «Microsoft Outlook» для составления расписания звонков или списка дел по работе с клиентами.

Предложенная технология осуществления изучаемых бизнес-процессов позволит: устранить существующие недостатки учета; управлять клиентской базой; улучшить взаимоотношения клиентами; повысить скорость и качество обработки заявок от клиентов; устранить ошибки, связанные с человеческим фактором; сделать работу «Отдела техподдержки» прозрачнее и т.п.

1.3. Постановка задачи

1.3.1. Цель и назначение автоматизированного варианта решения задачи

Автоматизации приема и анализа заявок отделом техподдержки ЧП «Net Soft System» позволит: вести учет сведений о клиентах; вести учет заявок, созданных пользователями; проводить контроль сроков исполнения заявок; проводить анализ занятости исполнителей; распределять заявки между исполнителями; создавать задачи; строить аналитические отчеты, позволяющие оценить работу каждого сотрудника «Отдела техподдержки» и т.п.

1.3.2. Общая характеристика организации решения задачи на ЭВМ

Проектируемая система должна состоять из трех модулей: «Пользователь», «Ответственный» и «Исполнитель».

В модуле «Пользователь» должны работать клиенты.

Модуль «Ответственный» создан для начальника «Отдела техподдержки».

Модуль «Исполнитель» предназначен системным администраторам, работающим в «Отделе техподдержки».

Все заявки должны иметь определённый статус, отражающий ход их исполнения. Анализ показал о целесообразности использования системы из пяти статусов.

Пользователи системы, работающие через модуль «Пользователь» имеют доступ к просмотру и изменению заявок, созданным только ими.

«Исполнители» могут просматривать заявки, в которых их имя указано в качестве исполнителя.

«Ответственный» имеет доступ ко всем заявкам, за исключением заявок, имеющим статус «Черновик».

Основными функциональными возможностями модуля «Пользователь» должны быть: создание заявки; отправка заявки на регистрацию; обмен сообщениями; обмен графическими материалами; смена логина и пароля.

Функция по созданию заявок предполагает создание в системе заявки, статус которой автоматически примет значение «Черновик».

Отправка заявки на регистрацию означает, что пользователь, заполнив заявку необходимыми сведениями, должен иметь возможность отправить ее на регистрацию.

Функция по обмену сообщениями предполагает, что пользователь может писать комментарии к созданной им заявке. При поступлении нового сообщения от исполнителя заявки, пользователю должно прийти уведомление о новых сообщениях.

«Обмен графическими материалами» предполагает возможность привязки к заявке изображений, скриншотов.

Смена логина и пароля позволит пользователю изменять параметры своей учетной записи.

Основными функциями модуля «Ответственный» должны быть: учет клиентов и пользователей; учет сотрудников; регистрация заявок; назначение исполнителя; смена исполнителя; продление срока исполнения заявки;

настройка приоритетов; ведение справочников; формирование и вывод на печать отчетов.

Функция «Учет клиентов» должна способствовать формированию единой клиентской базы.

Функция «Учет сотрудников» содержит сведения о работниках «Отдела техподдержки».

Функция «Регистрация заявок» означает, что при поступлении новых заявок на регистрацию, ответственный должен выбрать исполнителя заявки и присвоить заявке определенный номер. При регистрации заявки должны автоматически определяться дата регистрации и срок исполнения заявки (в соответствии с ее приоритетом).

Функция «Назначение исполнителя» предполагает выбор исполнителя заявки на основе информации о занятости каждого исполнителя (количество заявок в работе, количество заявок, ожидающих подтверждения).

Порой возникает необходимость сменить выбранного исполнителя и назначить нового, в связи с чем необходима функция «Смена исполнителя».

«Продление срока исполнения заявки» означает, что ответственный должен иметь возможность продлить срок исполнения заявки еще на 2 дня.

Функция по настройке приоритетов позволяет настроить количество дней, отводимых на исполнение заявок с тем или иным приоритетом.

Каждый из предусмотренных в системе справочников («ПО», «Сотрудники», «Темы заявок», «Приоритеты») должен в несколько раз ускорить работу пользователей за счет ввода заранее подготовленного набора значений.

Автоматическое формирование и вывод на печать отчетов предполагает, что система, на основе заранее заполненных справочников, и зарегистрированных пользователями данных должна автоматически формировать следующие виды отчетов: «Расписание задач»; «Отчет по

заявкам»; «Отчет по исполнителям»; «Отчет по клиентам»; «Отчет по просроченным заявкам»; «Расписание задач».

Основными функциями модуля «Исполнитель» должны быть: исполнение заявок; обмен сообщениями; обмен графическими материалами; создание задач; завершение работы над заявкой; смена логина и пароля.

Функция «Исполнение заявок» предполагает, что при выборе сотрудника исполнителем заявки, система должна тут же уведомить его об этом. Открыв заявку, исполнителю необходимо принять заявку на исполнение. В соответствии с данным действием меняется и статус заявки.

Функция «Обмен сообщениями» позволяет исполнителю обмениваться с пользователем комментариями для решения, поставленного в заявке вопроса.

Функция «Создание задач» позволит в рамках заявки создавать задачи и контролировать их исполнение.

Когда вопрос, поставленный в заявке, разрешен, исполнитель может завершить работу над заявкой. При этом статус заявки примет значение «Завершена».

Всем пользователям системы должна быть доступна возможность ведения поиска и фильтрации записей.

1.4. Анализ существующих разработок и обоснование выбора технологии проектирования

Современный рынок программных продуктов предлагает предприятиям множество готовых решений для работы с заявками. Наибольшей популярностью среди них пользуются такие системы, как «ServiceNow», «HelpDesk», «1C: ServiceDesk» и «WorkFlowSoft».

Для оценки особенностей перечисленных программных продуктов и соответствия их функциональных возможностей требованиям, предъявленным компанией ЧП «Net Soft System», произведем сравнительный анализ вышеуказанных систем (таблица 1).

Таблица 1. Сравнительная характеристика программных продуктов для учета заявок

Функциональные возможности	WorkFlowSoft	Service Desk	Service Now	Help Desk
Учет клиентов и пользователей	-	-	+	-
Создание заявок и отправка их на регистрацию	+	+	+	+
Назначение/изменение исполнителя заявки	-	-	+	+
Регистрация заявок с присвоением номера	+	+	+	+
Назначение сроков исполнения заявок согласно приоритетам	+	-	-	+
Настройка и продление сроков исполнения заявок	-	-	+	-
Обмен сообщениями	+	+	+	+
Обмен графическими материалами	+	+	+	+
Автоматическая смена статуса заявки	+	+	-	+
Уведомления о новых сообщениях и заявках	-	-	+	+
Ведение справочников	-	-	+	+
Формирование и вывод на печать отчетов	+	+	+	+
Поиск и фильтрация записей	+	+	+	+
Интуитивно-понятный пользовательский интерфейс	+	-	+	+

Анализ приведенных в таблице данных свидетельствует, что все перечисленные системы обладают минимальным набором функций, необходимых отделу техподдержки ЧП «Net Soft System» для работы с заявками.

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

Программный продукт «WorkFlowSoft» от компании «WorkFlowSoft» не соответствует всем предъявленным требованиям к функциональности, но имеет невысокую стоимость (350 рублей на пользователя в месяц) [Ошибка! Источник ссылки не найден.Ошибка! Источник ссылки не найден.], [Ошибка! Источник ссылки не найден.Ошибка! Источник ссылки не найден.].

В продукте компании «1С» - «ServiceDesk» отсутствуют базовый набор функций, необходимых для полноценного учета заявок. Интерфейс программы имеет сложную и непонятную структуру. Стоимость решения будет формироваться из зарплаты программиста (где-то от 20000 до 40000 руб. в месяц) [14].

«ServiceNow» от компании «ITSM365» является серьезным решением с большим набором функций. Версия программы, рассчитанная на 10 лицензий, обойдется 20000 рублей в месяц [Ошибка! Источник ссылки не найден.].

Программа «HelpDesk» от компании «IT GUILT» обойдется от 5 до 25 тысяч рублей в месяц в зависимости от количества пользователей и регистрируемых заявок [Ошибка! Источник ссылки не найден.].

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

Обычно есть два выхода из таких ситуаций: первый – подогнать бизнес-процессы компании под требования системы; второй – доработать внедряемую систему под нужды компании.

Во втором случае есть риск, что после доработки проекта, готовое решение не будет приносить желаемый результат в связи с растущими

требованиями бизнеса. К тому же внедрение готовых решений не гарантирует обеспечение безопасности данных, т.к. они находятся на стороннем сервере и имеется вероятность взлома и потери личных данных клиентов.

Итоги анализа существующих на отечественном рынке программных продуктов для отдела техподдержки доказывают, что наиболее рациональным решением для ЧП «Net Soft System» является разработка собственной информационной системы, специально ориентированной под ее бизнес-процессы. Кроме этого большое значение сыграл тот фактор, что у данной организации имеются свои разработчики и программисты, которые могут в любое время изменить или добавить любой функционал к данной программе.

Выводы по главе 1

Подводя итоги первой главы, где был проведен анализ деятельности ЧП «Net Soft System». Было выявлено, что одним из важных подразделений компании является «Отдел техподдержки», основная задача которых - принимать и исполнять заявки, поступающие от заказчиков.

Описание существующей технологии учета заявок позволило выявить ряд недостатков, связанных с отсутствием на предприятии системы автоматизированного учета деятельности «Отдела техподдержки».

Для устранения существующих недостатков учета заявок, была разработана новая технология осуществления информационно-технической поддержки пользователей, основная суть которой состоит в применении информационной системы учета заявок.

Существующие аналоги не соответствуют заявленным требованиям и сохранности личных данных клиентской базы ЧП «Net Soft System».

Кроме этого, учитывая все аспекты по дальнейшему развитию и расширению возможностей предприятия, а также возможностей добавления любых вспомогательных функций и компонентов для автоматизации бизнес

процессов, руководством было принято решение о создании собственного программного продукта.

Глава 2. Разработка и реализация проектных решений

2.1. Логическое моделирование предметной области

2.1.1. Логическая модель и ее описание

В современном проектировании информационных систем широко используется визуальное моделирование при помощи унифицированного языка моделирования UML.

Унифицированный язык моделирования является графическим языком для визуализации и конструирования, и систем.

Основные диаграммы UML: вариантов использования; классов; кооперации; последовательности; состояний; деятельности; компонентов; развертывания [8].

Диаграмма вариантов использования (диаграмма прецедентов) - это визуальное изображение, отношений между актёрами и вариантами использования (рисунок 11).

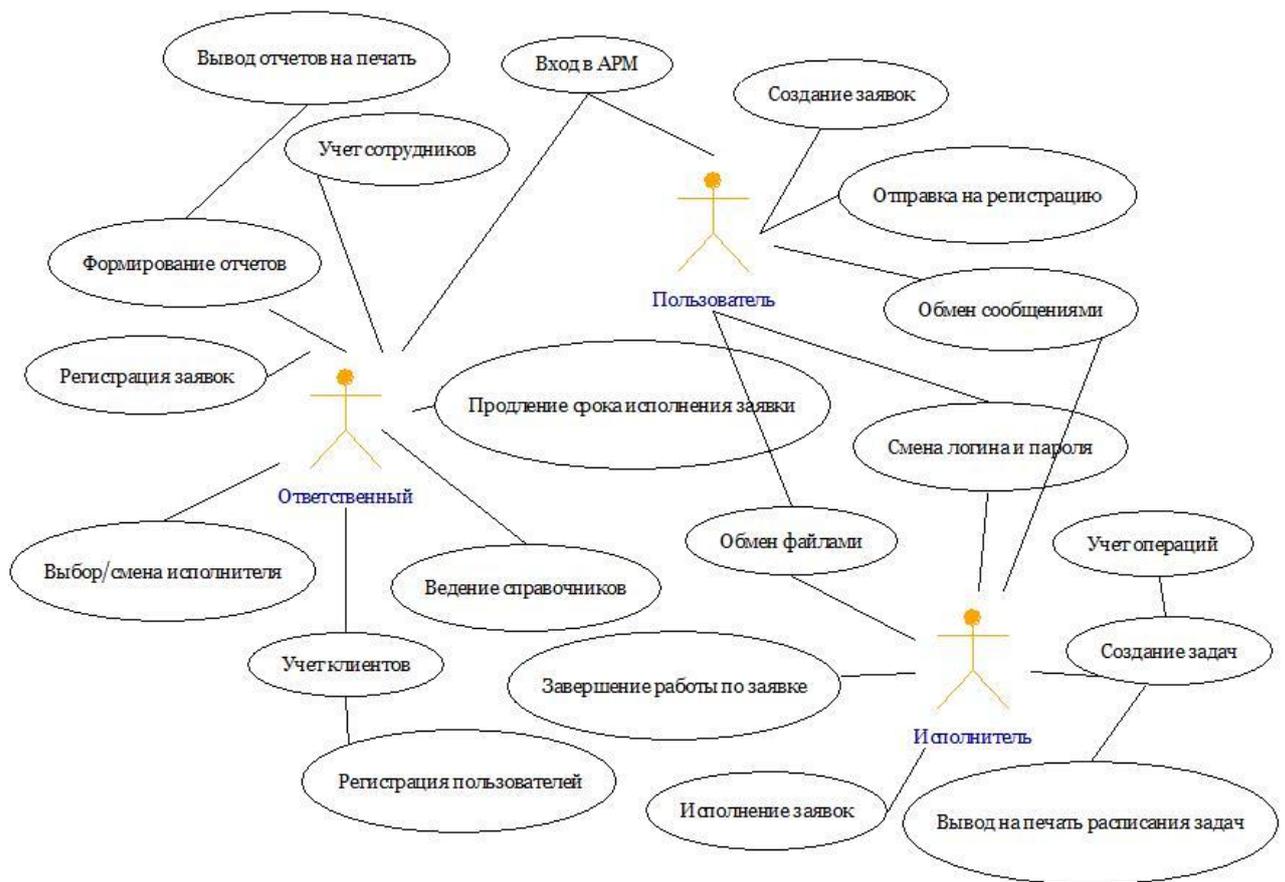


Рисунок 11. Диаграмма вариантов использования

Основные элементы нотации диаграмм деятельности: деятельности; переходы; элементы выбора; синхронизации.

Деятельностью называется исполнение определенного поведения в потоке управления системы (рисунок 12).



Рисунок 12. Диаграмма деятельности

Диаграмма состояний отображает нахождение одиночного объекта, события или сообщения, которые инициируют переход из одного состояния в другое, и действия, как результат смены его состояния (рисунок 13).

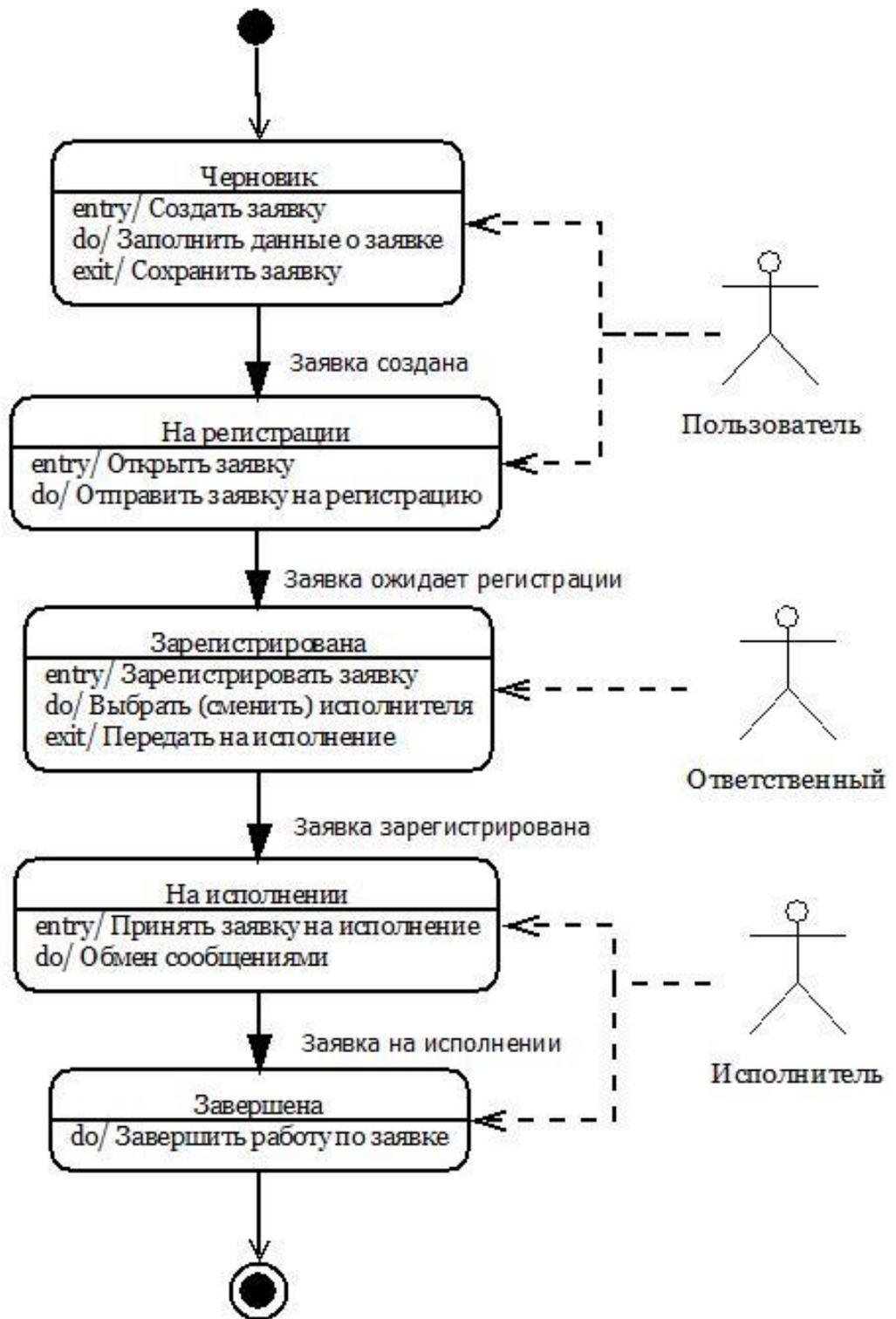


Рисунок 13. Диаграмма состояний

Диаграмма последовательности действий отображает связь между объектами, упорядоченную по времени. Линия жизни объекта отражает существование объекта в течение определенного периода времени (рисунок 14).

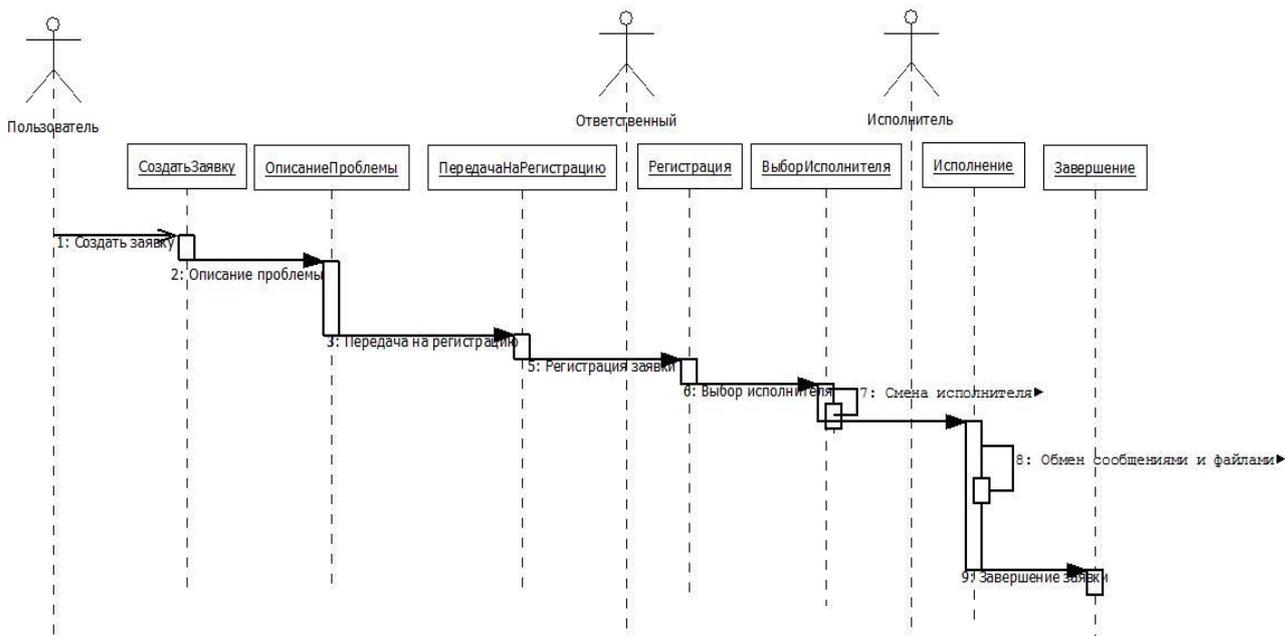


Рисунок 14. Диаграмма последовательности

Процесс создания структуры базы данных, является многоуровневым, в результате, которого осуществляется переход от предметного рассмотрения к конкретной реализации базы данных средствами соответствующей СУБД: инфологическая модель предметной области, логическая модель данных, физическая модель данных [10 **Ошибка! Источник ссылки не найден.Ошибка! Источник ссылки не найден.**].

Инфологической моделью предметной области являются наши понятия о предметной области и логических связях между данными, выраженные при помощи специализированных графических средств. В качестве такого графического средства, выбран свободный редактор диаграмм «Visio 2016».

Инфологическая модель разрабатываемой информационной системы приведена на рисунке 15.

Построенная ранее инфологическая модель служит основой для разработки логической модели данных.

Одним из подходов к построению модели базы данных является методология «IDEF 1X», основанная на концепции «сущность-связь».

Сущность на диаграмме обозначается в виде прямоугольника, разделенного на две части:

- ключевая область, представленная ключевыми атрибутами;
- область данных, состоящую из не ключевых атрибутов.

Ключевыми считаются атрибуты, имеющие первичный ключ. Все остальные атрибуты относятся к не ключевым (в том числе и атрибуты, составляющие внешний ключ [3].

Логическая модель данных бизнес-процесса «Информационно-техническое обслуживание пользователей» на основе методологии «IDEF 1X» представлена на рисунке 16.

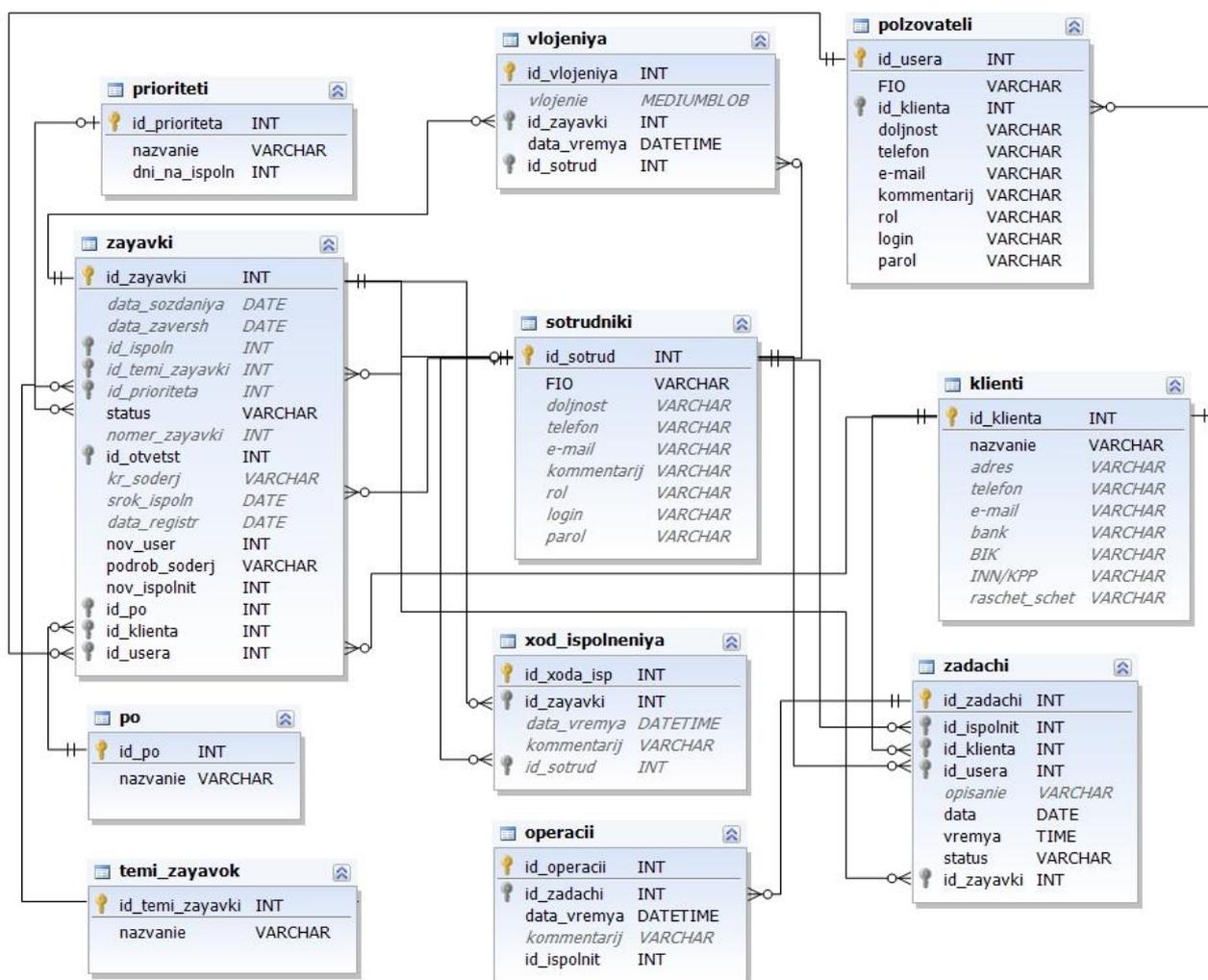


Рисунок 16. Логическая модель данных

Модель базы данных, изображенная на рисунке 22, показывает, что проектируемая БД состоит из 11 таблиц, детальная информация о которых будет приведена при описании физической модели данных.

2.1.2. Используемые классификаторы и системы кодирования

Классификатор – перечень наименованных объектов, каждому из которых присвоен уникальный код [10].

В системе приема и анализа заявок отделом техподдержки ООО «Net Soft System» предполагается использовать следующие классификаторы и системы кодирования (таблица 2):

Таблица 2. Классификаторы и системы кодирования

Кодируемое множество	Длина кода	Мощность кода	Система кодирования
Код клиента	3 знака	999 вариантов	Порядковая
Код пользователя	4 знака	9999 вариантов	Порядковая
Код сотрудника	4 знака	9999 вариантов	Порядковая
Код названия ПО	3 знака	999 вариантов	Порядковая
Код темы заявки	3 знака	999 вариантов	Порядковая
Код приоритета	1 знак	3 вида приоритета	Порядковая
Код заявки	5 знаков	99999 вариантов	Порядковая
Статус заявки	1 знак	5 статусов заявок	Порядковая
Статус задачи	1 знак	3 статуса задачи	Порядковая

2.1.3. Характеристика нормативно-справочной и входной оперативной информации

Входная информация – это данные поступающие в систему извне. Для изучаемого бизнес-процесса такими источниками входных данных могут служить справочники системы: «ПО», «Сотрудники», «Темы заявок», «Приоритеты».

На основе информации из перечисленных справочников осуществляются все процессы по учету заявок отделом техподдержки. Регистрация таких операций производится в отдельных таблицах, также служащих в качестве входных источников информации: «Клиенты», «Пользователи», «Заявки», «Ход исполнения», «Вложения», «Задачи» и «Операции».

Рассмотрим назначение каждого из перечисленных ранее справочников и таблиц.

Справочник «ПО» необходим для ведения перечня программных продуктов, разработанных компанией.

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

Справочник «Приоритеты» позволит управлять сроками исполнения заявок, отведенными на каждый вид приоритета.

Справочник «Сотрудники» служит для хранения информации о сотрудниках компании: ФИО, роль, клиент, телефон, E-mail и данных для входа в систему (логин и пароль).

Таблица «Клиенты» служит для хранения информации о клиентах компании: название, телефон, E-mail, банковские реквизиты, пользователи и т.п.

Таблицы «Заявки», «Ход исполнения», «Вложения», «Задачи» и «Операции» необходимы для регистрации заявок в системе, их исполнения, прикрепления различных графических материалов, создания задач и контроля их исполнения.

Таблица «Заявки» позволяет вести учет заявок в системе.

Таблица «Ход исполнения» позволяет регистрировать действия, осуществляемые над заявками и обмениваться сообщениями.

Таблица «Вложения» необходима для учета графических материалов, которыми обмениваются пользователь и исполнитель в ходе исполнения заявок.

Таблица «Задачи» позволяет создавать список дел, которые необходимо осуществить в срок.

Таблица «Операции» необходима для учета и контроля исполнения задач. Для регистрации операций по задаче используются два макета: «Набор номера» и «Отправка сообщения».

2.1.4. Физическая модель базы данных

Физические модели данных информационных систем представляют собой логические модели данных, выраженных средствами конкретной СУБД. В качестве средства разработки БД будет использована реляционная система управления БД MySQL.

При создании физических моделей данных отношения, описанные в логической модели, преобразуются в таблицы, атрибуты становятся столбцами, каждому атрибуту присваивается тип данных, а для ключевых столбцов создаются уникальные индексы [3].

Как было отмечено ранее, проектируемая БД состоит из 11 таблиц, перечень и структура которых представлены ниже.

Таблица 3. Атрибуты сущности «ПО»

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

Таблица 4. Атрибуты сущности «Клиенты»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_klienta	int(10) unsigned	NO	PRI	auto_increm.	Код клиента
nazvanie	varchar(65)	NO			Название
adres	varchar(100)	YES			Адрес
telefon	varchar(20)	NO			Телефон
e-mail	varchar(20)	YES			Е-mail
bank	varchar(45)	YES			Банк
BIK	varchar(25)	YES			БИК
INN/KPP	varchar(25)	YES			ИНН/КПП
raschet_schet	varchar(25)	YES			Расчетный счет

Таблица 5. Атрибуты сущности «Приоритеты»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_prioriteta	int(10) unsigned	NO	PRI	auto_increm.	Код приоритета
nazvanie	varchar(20)	NO			Название
dni_na_ishpoln	int(10) unsigned	NO		0	Кол-во дней на испол-е

Таблица 6. Атрибуты сущности «Пользователи»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_usera	int(10) unsigned	NO	PRI	auto_increm.	Код пользователя
FIO	varchar(45)	NO			ФИО
id_klienta	int(10) unsigned	YES	MUL		Код клиента
doljnost	varchar(145)	YES			Должность
telefon	varchar(20)	YES			Телефон
e-mail	varchar(20)	YES			E-mail
rol	varchar(25)	YES			Роль
login	varchar(45)	YES			Логин
parol	varchar(45)	YES			Пароль

Таблица 7. Атрибуты сущности «Сотрудники»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_sotrud	int(10) unsigned	NO	PRI	auto_increm.	Код пользователя
FIO	varchar(45)	NO			ФИО
doljnost	varchar(145)	YES			Должность
telefon	varchar(20)	YES			Телефон
e-mail	varchar(20)	YES			E-mail
rol	varchar(25)	YES			Роль
login	varchar(45)	YES			Логин
parol	varchar(45)	YES			Пароль

Таблица 8. Атрибуты сущности «Темы заявок»

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

Таблица 9. Атрибуты сущности «Вложения»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_vlojeniya	int(10) unsigned	NO	PRI	auto_increm.	Код вложения
vlojenie	mediumblob	YES		(null)	Вложение
id_zayavki	int(10) unsigned	NO	MUL	0	Код заявки
data_vremya	datetime	NO		0000-00-00 00:00:00	Дата и время
id_sotrud	int(10) unsigned	NO	MUL	0	Код пользователя

Таблица 10. Атрибуты сущности «Ход исполнения»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_xoda_isp	int(10) unsigned	NO	PRI	auto_increm.	Код хода исполнения
id_zayavki	int(10) unsigned	NO	MUL	0	Код заявки
data_vremya	datetime	YES			Дата и время
kommentarij	varchar(600)	YES			Комментарий
id_sotrud	int(10) unsigned	YES	MUL		Код пользователя

Таблица 11. Атрибуты сущности «Заявки»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_zayavki	int(10) unsigned	NO	PRI	auto_increm.	Код заявки
id_usera	int(10) unsigned	NO	MUL	0	Код пользователя
data_sozdaniya	date	YES			Дата создания
data_zaversh	date	YES			Дата завершения
id_ispoln	int(10) unsigned	YES	MUL		Код исполнителя
id_temi_zayavki	int(10) unsigned	YES	MUL		Код темы заявки
id_prioriteta	int(10) unsigned	YES	MUL		Код приоритета
status	varchar(15)	NO		Черновик	Статус
nomer_zayavki	int(10) unsigned	YES			Номер заявки

Продолжение таблицы 11

id_otvetst	int(10) unsigned	NO	MUL	0	Код ответственного
kr_soderj	varchar(100)	YES			Краткое содержание
srok_ispoln	date	YES			Срок исполнения
data_registr	date	YES			Дата регистрации
nov_user	int(10) unsigned	NO		0	Новое сообщение1
podrob_soderj	varchar(245)	NO			Подробное содержание
nov_ispolnit	int(10) unsigned	NO		0	Новое сообщение 2
id_po	int(10) unsigned	NO		0	Код ПО
id_klienta	int(10) unsigned	NO	MUL	0	Код клиента

Таблица 12. Атрибуты сущности «Задачи»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_zadachi	int(10) unsigned	NO	PRI	auto_increm.	Код задачи
id_ispolnit	int(10) unsigned	NO	MUL	0	Код исполнителя
id_klienta	int(10) unsigned	NO	MUL	0	Код клиента
id_usera	int(10) unsigned	NO	MUL	0	Код пользователя
opisanie	varchar(205)	YES			Описание
data	date	NO		0000-00-00	Дата
vremya	time	NO		0:00:00	Время
status	varchar(15)	NO			Статус
id_zayavki	int(10) unsigned	NO	MUL	0	Код заявки

Таблица 13. Атрибуты сущности «Операции»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_operacii	int(10) unsigned	NO	PRI	auto_increm.	Код операции
id_zadachi	int(10) unsigned	NO	MUL	0	Код задачи
data_vremya	datetime	NO		0000-00-00 00:00:00	Дата/время
kommentarij	varchar(245)	YES			Комментарий
id_ispolnit	int(10) unsigned	NO		0	Код исполнителя

Описание обозначений базы данных представлено в приложении 1.

2.1.5. Характеристика результатной информации

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

Основными выходными документами проектируемой системы являются отчеты: «Расписание задач»; «Отчет по заявкам»; «Отчет по исполнителям»; «Отчет по клиентам»; «Отчет по просроченным заявкам».

«Расписание задач» отражает перечень задач исполнителя на определенную дату.

«Отчет по заявкам» должен отражать перечень зарегистрированных в системе заявок за определенный временной интервал.

«Отчет по исполнителям» должен отражать количество заявок, в которых принимал участие каждый исполнитель, а также выдавать общее количество этих заявок.

«Отчет по клиентам» имеет аналогичное значение, но касается не исполнителей, а пользователей системы (клиентов).

«Отчет по просроченным заявкам» отражает перечень заявок, где «Дата завершения» заявки выше отведенного «Срока исполнения». Такой отчет позволяет вести контроль сроков исполнения заявок

В рамках данного параграфа: был определен перечень входных и выходных данных разрабатываемой ИС; построена модель данных на инфологическом, логическом и физическом уровнях.

2.2. Физическое моделирование информационной системы

2.2.1. Выбор архитектуры информационной системы

Выбор архитектуры, разрабатываемой ИС определяется задачами, для решения которых она создается. В соответствии с этим, программа должна иметь многопользовательскую, клиент-серверную архитектуру, которая предполагает, что пользователи будут работать с единой базой данных, расположенной на удаленном сервере.

База данных управляемая СУБД является основной формой для осуществления организации информации на машинных носителях.

Для выбора системы управления базами данных необходимо провести сравнительный анализ наиболее популярных СУБД (MS Access, Visual FoxPro, MySQL и Oracle) на предмет соответствия основным функциям, которые она должна выполнять.

В таблице 14 приводится сравнение возможностей, рассматриваемых СУБД на предмет использования в проектируемой ИС.

Таблица 14. Сравнительный анализ СУБД

Оцениваемое средство	PostgreSQL	MySQL	Oracle
Команды формирования SQL-запросов	+	+	+
Обработка сбойных ситуаций	+	+	+
Администрирование БД	+	+	+
Копирование и восстановление БД	+	+	+
Взаимодействие с другими приложениями	+-	+	+-
Сложность освоения	-	+	-
Скорость работы	+	+	+
Возможность некоммерческого использования	+	+	-

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

MySQL – СУБД, полностью удовлетворяющая требованиям безопасности, надежности, компактного хранения и скорости обработки данных с мощными средствами администрирования БД.

Выбор среды разработки будет производиться на основе анализа различных версии таких систем программирования как: Embarcadero RAD Studio XE. В таблице 15 приводится экспертная оценка возможностей рассматриваемых систем.

Таблица 15. Экспертная оценка средств разработки приложений версий

Embarcadero RAD Studio XE

Возможности	RAD Studio XE Architect	RAD Studio XE Enterprise	RAD Studio XE Professional
имеется подключение к локальным базам данных InterBase, MySQL в Delphi и C++Builder	+	+	+
имеет подключение к базам данных при помощи ADO.NET к локальным базам данных InterBase в Delphi Prism	+	+	+
библиотека визуальных компонентов (VCL) для веб-решений (ограниченная функциональность) с поддержкой только автономных приложений и не более 5 подключений	+	+	+
поддержка ISAPI и HTTPS отсутствует	+	+	+
набор метрик и средств аудита кода начального	+	+	+

уровня, помогающих проанализировать стиль программирования и состояние проекта			
визуализация кода с использованием универсального языка моделирования (UML), упрощающего анализ исходного кода благодаря визуальному представлению	+	+	+
имеет подключение к серверам баз данных InterBase, Firebird, MySQL, Microsoft SQL Server, Oracle, DB2, Informix и Sybase при релизации приложений с собственным кодом Windows	+	+	-
имеет подключение к удаленным серверам InterBase при создании приложений .NET	+	+	-
присутствует разработка многозвенных приложений Data Snap и разработка клиентов .NET Data Snap	+	+	-

Продолжение таблицы 15.

Final Builder Embarcadero Edition и средства командной строки для автоматизации сборок	+	+	-
библиотека визуальных компонентов (VCL) для веб-решений с поддержкой автономных приложений, служб, расширений ISAPI, HTTPS и неограниченного числа подключений	+	+	-
полный набор, включающий более 80 метрик и более 200 показателей, используемых для аудита кода	+	+	-
полное двустороннее UML-моделирование с несколькими типами диаграмм, синхронизацией кода и моделей, созданием документации и шаблонами разработки	+	+	-
обратное проектирование, анализ и оптимизация баз данных	+	-	-
имеет возможность автоматического	+	-	-

создания кода баз данных на основе моделей для прямого проектирования			
создание логических и физических моделей на основе данных, извлеченных из баз, данных и файлов сценариев	+	-	-
работа с удобными для чтения и навигации диаграммами	+	-	-
усовершенствованное сравнение и объединение с помощью двунаправленных операций сравнения и объединения моделей и структур баз данных	+	-	-

Программный комплекс Embarcadero RAD Studio — это один из мощнейших комплексов создания, обновления и взаимодействия базы данных с визуально профессиональной оболочкой и пользовательским интерфейсом для Windows 10, Mac, мобильных устройств, IoT и других платформ при помощи Object Pascal и C++. Благодаря широкому выбору функций поддержки Windows 10, в том числе новых компонентов VCL для Windows 10, стилей для VCL и FMX, а также службы UWP (универсальной платформы Windows), например, уведомления, даёт возможность с легкостью и быстротой осуществить перенос ваших приложения в Windows 10, при этом сохраняя текущих пользователей.

Данная платформа даёт возможность для поддержки более крупных проектов на значительно большем количестве платформ с увеличенным вдвое объемом памяти в среде разработки и вдвойне большим размером поддерживаемых проектов. Платформа осуществляет поддержку нескольких мониторов, а также имеет большое количество новых функций среды разработки, предназначенных для ускорения создания кода, которую делают работу на ней значительно приятней и эффективней. При создании приложений, благодаря RAD Studio 10 у разработчиков в 5 раз увеличилась скорость создание приложений, по сравнению с иными инструментами.

Таким образом, разработка приложений для нескольких настольных, мобильных, облачных платформ и платформ баз данных, включая 32- и 64-разрядные версии Windows 10, Mac OS X, iOS и Android, происходит еще быстрее.

В наше время для независимых и корпоративных разработчиков задачей №1 является, поддержка быстрорастущей совокупности в различных устройствах Windows и Apple, типов пользовательских интерфейсов, версий ОС и разнородных форм-факторов. Для преобразования, указанного в конкурентные преимущества, компаниям необходимо найти возможность значительно нарастить силы и ресурсы сотрудников отдела разработки, при этом всё осуществить за счет технологических решений.

Программный комплекс Embarcadero RAD Studio XE10 и есть тот самый «мультипликатор производительности» разработчиков, позволяющий быстро и с применением визуальных методов буквально ставить на поток создание приложений для Windows 8, Mac, .NET, Web и мобильных платформ. В одном «пакете» вы получаете Delphi®, C++Builder®, Embarcadero Prism™ и HTML5 Builder, поэтому скорость разработки минимум в 5 раз, охватывая Windows 8, Mac OS X с поддержкой дисплеев Retina и многое другое.

Это предоставляет возможность существенно сократить время выпуска на рынок новых продуктов, и получить при этом значительные конкурентные преимущества и перспективы в направлении развития бизнеса, благодаря мульти платформенным визуальным средствам разработки на компонентной основе и с возможностью эффективного взаимодействия с различными базами данных и облачными сервисами.

По результатам экспертного анализа для разработки, мы будем использовать уже имеющееся в нашей организации ПО Embarcadero RAD Studio XE 10.2.2 Tokyo Architect.

Программный комплекс Embarcadero RAD Studio XE10 Architect

предназначен для разработчиков, реализующих традиционные клиент/серверные приложения, интенсивно взаимодействующие с базами данных и облачными сервисами, а также Web-приложения в рамках корпоративных систем.

RAD Studio Architect сочетает в себе технологию быстрой разработки приложений с возможностями визуального моделирования данных, что помогает разработчикам проектировать наиболее критичные структуры данных в корпоративной системе. RAD Studio Architect включает всё, что есть в редакции Enterprise, а также мощные средства для моделирования и проектирования баз данных:

- Обратное проектирование (reverse-engineer), анализу и оптимизации баз данных
- Автоматическая генерация кода для создания баз данных на основе моделей для прямого проектирования (forward-engineering)
- Создание логической и физической модели на основании информации из базы данных и файлов со скриптами
- Генерация ясных и понятных диаграмм с возможностью лёгкой навигации
- Продвинутое возможности сравнения и слияния с двунаправленным процессом сравнения и слияния структур моделей и баз данных

Все выше перечисленные факторы сыграли значимую роль при выборе руководством, программного обеспечения Embarcadero RAD Studio XE 10.2.2 Tokyo Architect для использования разработчиками и программистами предприятия ЧП «Net Soft System».

2.2.2. Функциональная схема проекта

Разработанная программа состоит из 41 формы. Основные функциональные возможности системы представлены на рисунке 17 в виде дерева функций.

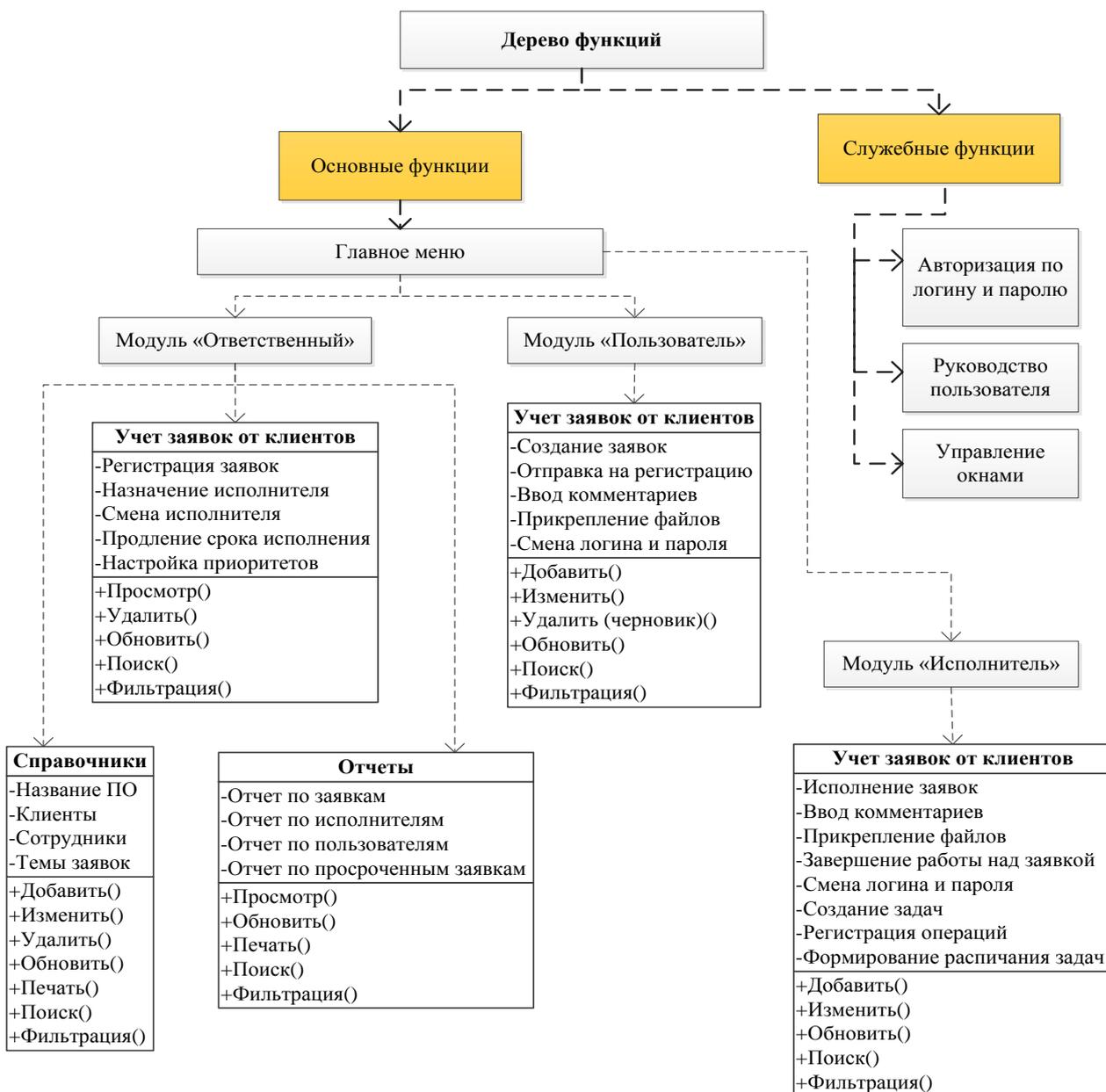


Рисунок 17. Дерево функций

2.2.3. Описание алгоритма идентификации пользователей

Согласно структурной схеме пакета, для входа в АИС необходимо пройти процедуру авторизации. Каждый пользователь системы имеет свой уникальный логин и пароль для доступа к программе.

Описание алгоритма идентификации пользователя представлено на рисунке 18.

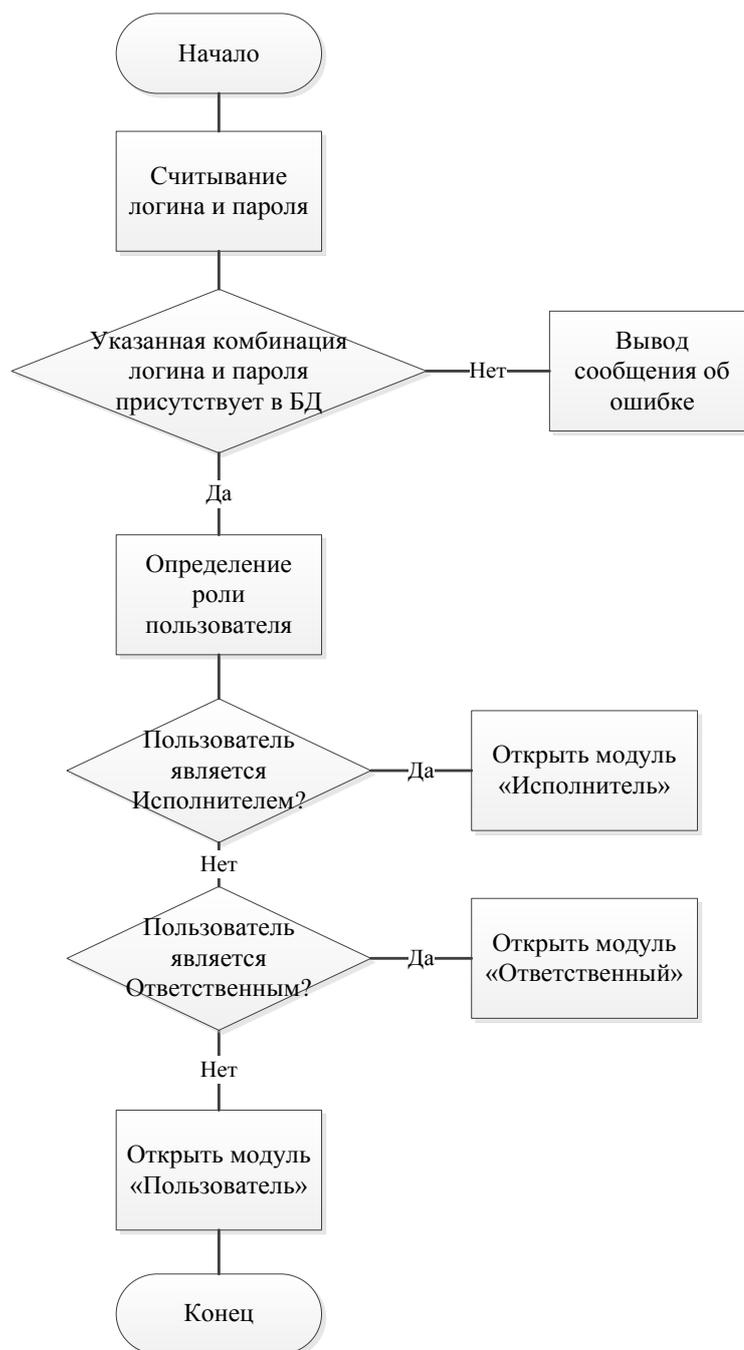


Рисунок 18. Алгоритм идентификации пользователя

2.2.4. Схема взаимосвязи программных модулей и информационных файлов

Схема взаимосвязи программных модулей и информационных файлов отражает взаимосвязь между программным и информационным обеспечением задачи.

Рассмотрим схемы взаимодействия программных модулей и информационных файлов при выполнении следующих функций АИС:

- проверка прав доступа к программе;
- обработка заявок;
- формирование отчетов.

Схема взаимодействия программных модулей и информационных файлов при проверке прав доступа к программе представлена на рисунке 19.



Рисунок 19. Схема взаимодействия программных модулей и информационных файлов при проверке прав доступа к программе

Схема взаимодействия программных модулей и информационных файлов при обработке заявок представлена на рисунке 20.

Схема взаимодействия программных модулей и информационных файлов при формировании отчета по заявкам представлена на рисунке 21.

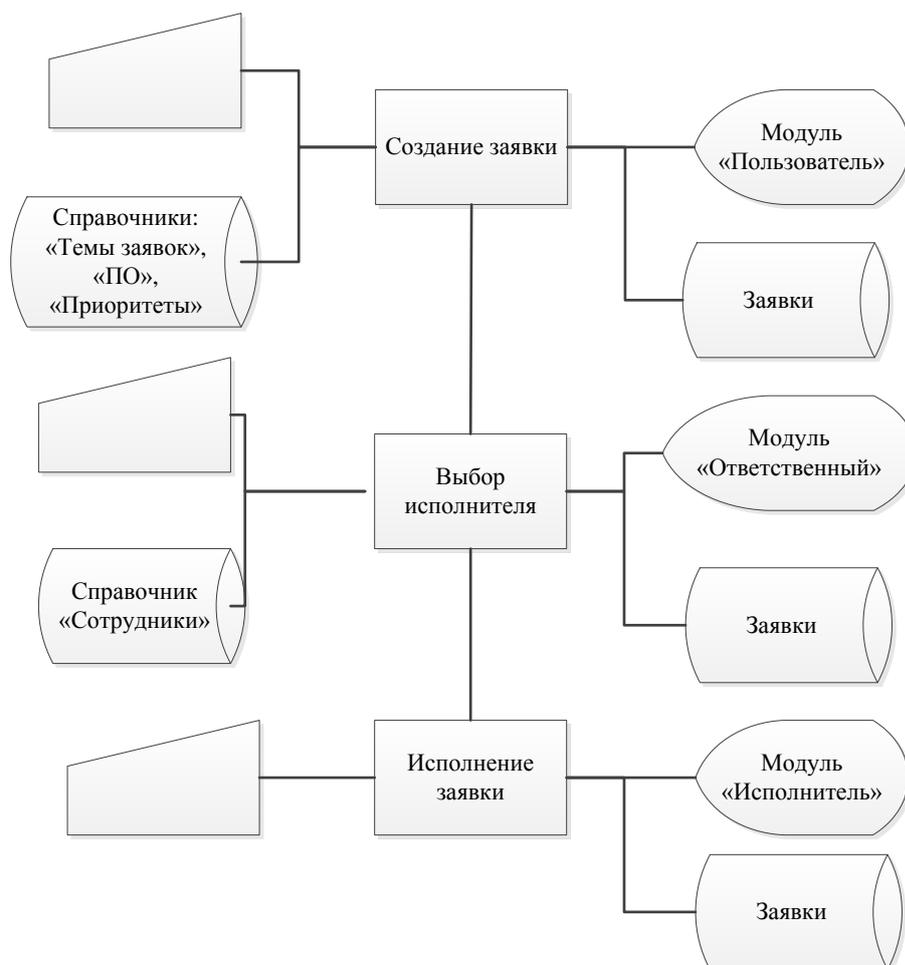


Рисунок 20. Схема взаимодействия программных модулей и информационных файлов при обработке заявок



Рисунок 21. Схема взаимодействия программных модулей и информационных файлов при формировании отчета по заявкам

2.3. Технологическое обеспечение задачи

2.3.1. Организация технологии сбора, передачи, обработки и выдачи информации

При запуске информационной системы приема и анализа заявок отделом техподдержки ЧП «Net Soft System» появляется заставка с названием программы. Длительность отображения заставки 1 секунда, после чего открывается окно «Вход в систему» для ввода имени пользователя и соответствующего ему пароля.

После успешной авторизации в системе, пользователь, в зависимости от своей должности, получит доступ к одному из двух модулей программы: «Ответственный» или «Пользователь» или «Исполнитель».

Особенности работы с каждым модулем программы представлены в приложении 3.

2.3.2. Схема технологического процесса сбора, передачи, обработки и выдачи информации

Рассмотрим схему технологического процесса для модуля «Ответственный». Работа с разработанной информационной системой начинается с входа в программу и выбора соответствующего пункта меню. Для входа в личный кабинет необходимо выбрать пункт меню «Личный кабинет» (А); для работы со справочниками – пункт меню «Справочники» (Б); для просмотра и формирования отчетов – пункт меню «Отчеты» (В); для получения справочной информации – пункт меню «Справка» (Г). Технологическая схема входа в ИС представлена на рисунке 22.

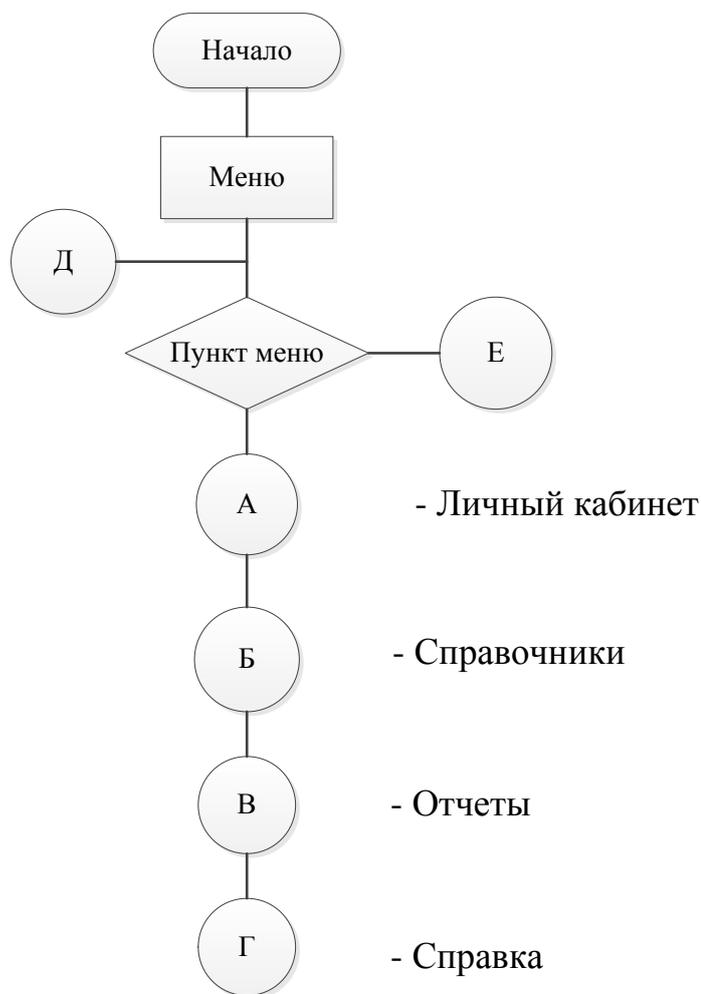


Рисунок 22. Технологическая схема входа в ИС

После выбора соответствующего раздела меню, происходит переход в пункты меню этого раздела. Пользователь может выбрать один из представленных пунктов меню выбранного раздела, либо вернуться к выбору другого раздела меню.

Технологическая схема выбор пункта меню «Личный кабинет» представлена на рисунке 23.

Выбрав один из пунктов меню («Заявки» либо «Настройка приоритетов»), пользователь может выполнить над ним следующие действия (добавить запись, изменить текущую запись, удалить запись, обновить, отфильтровать записи). Технологическая схема выбора действий, выполняемых над пунктом меню А1, представлена на рисунке 24.

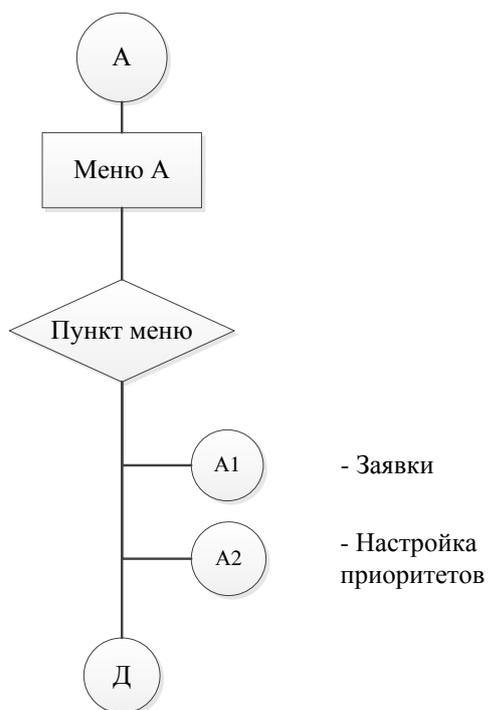


Рисунок 23. Технологическая схема выбор пункта меню «Личный кабинет»

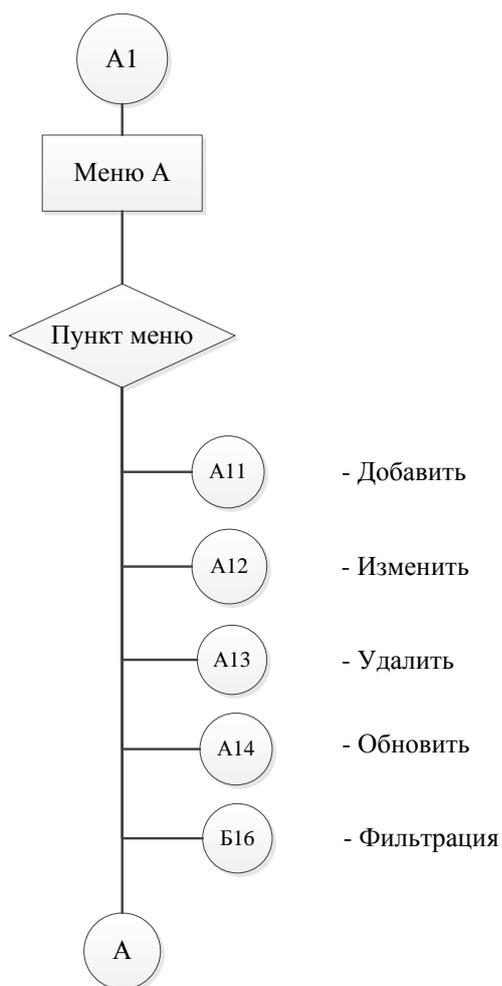


Рисунок 24. Технологическая схема выбора действий

Технологическая схема выбор пункта меню «Справочники» представлена на рисунке 25.

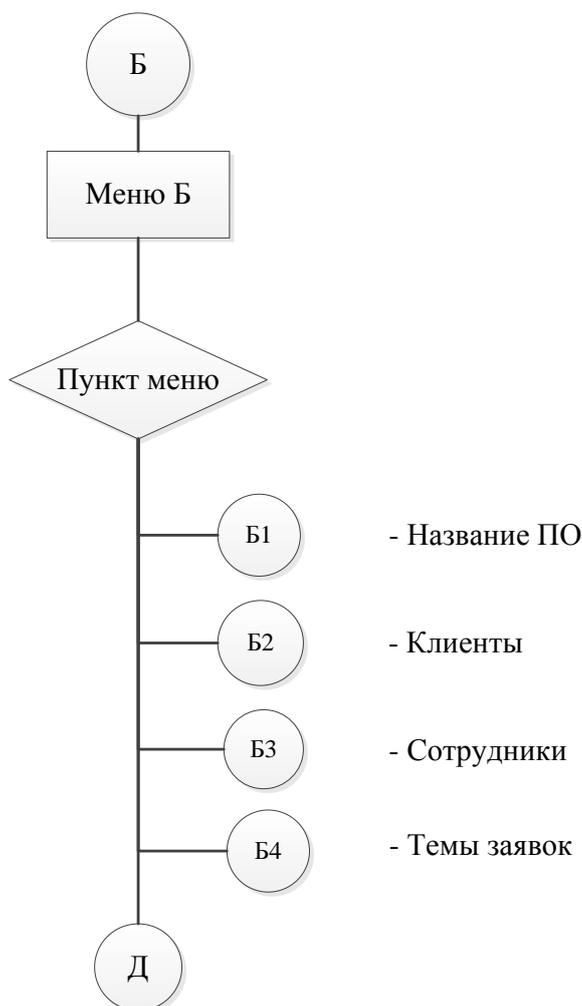


Рисунок 25. Технологическая схема выбор пункта меню «Справочники»

Выбрав работу с одним из справочников, пользователь может выполнить над ним следующие действия (добавить запись в справочник, изменить текущую запись, удалить запись, обновить содержимое справочника, произвести поиск по определённому значению, отфильтровать записи).

Технологическая схема выбора действий, выполняемых над содержимым справочника, представлена на рисунке 26.

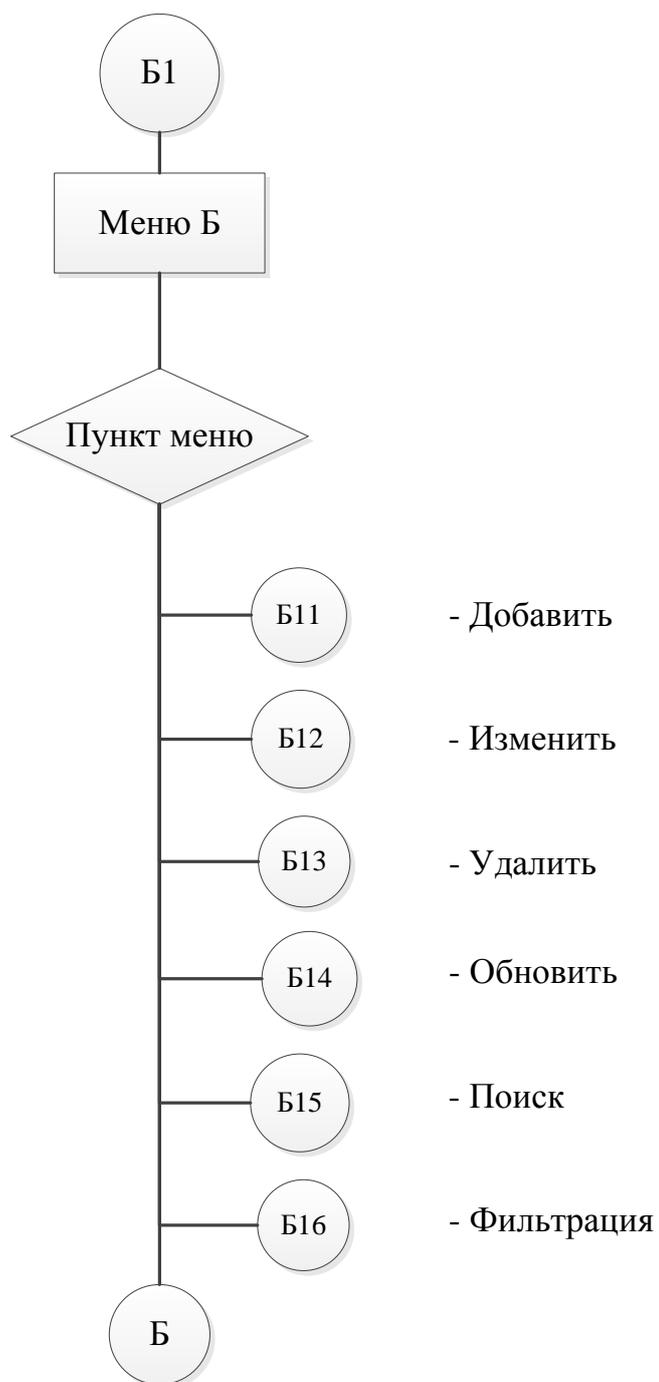


Рисунок 26. Технологическая схема выбора действий, выполняемых над содержимым справочника

Выводы по главе 2

В рамках второй главы:

- было произведено моделирование системы с использованием языка UML;
- определены входные источники информации, которыми служат справочники компании, и описаны выходные документы – отчеты.
- произведен выбор средств разработки. В качестве средства разработки приложения была выбрана среда объектно-ориентированного программирования Embarcadero RAD Studio XE10 Architect Tokyo ver.10.2.2, а в качестве средства разработки базы данных – система управления базами данных MySQL.
- разработана логическая и физическая модель БД, которая показала, что проектируемая БД состоит из 11 таблиц;
- разработана информационная система, способная решать следующие задачи: учет клиентов и пользователей; создание заявок и отправка их на регистрацию; назначение/изменение исполнителя заявки; регистрация заявок с присвоением номера; обработка и исполнение заявок; назначение сроков исполнения заявок согласно приоритетам; настройка и продление сроков исполнения заявок; обмен сообщениями; обмен графическими материалами; автоматическая смена статуса заявки; уведомление о новых сообщениях и заявках; формирование расписания задач и контроль их исполнения; учет сотрудников; ведение справочников; формирование отчетов; поиск и фильтрация записей; многопользовательский режим работы и т.д.

Глава 3. Оценка и обоснование экономической эффективности проекта

3.1. Планирование и контроль выполнения работ

Для организации и планирования работ по разработке и внедрению программного продукта, необходимо объединить работы в этапы. Это позволит отслеживать стадии выполнения проекта, контролировать степень завершенности работ и проекта в целом.

Выделенные этапы проектирования и разработки системы с оценкой их продолжительности представлены в таблице 16.

Таблица 16. Этапы разработки системы

№ этапа	Наименование планируемых работ	Количество дней по плану
1	Формирование требований ПО	9
2	Разработка концепции ПО	7
3	Разработка технического задания ПО	9
4	Разработка серверной части ПО	16
5	Разработка клиентской части ПО	19
6	Тестирование и отладка системы ПО	14
7	Написание сопроводительной документации	7
8	Внедрение системы	9
Итого:		90

Общий срок разработки системы составит 90 дней.

Состав участников проекта будет следующий: руководитель проекта – штатный сотрудник, начальник отдела; программист – штатный сотрудник отдела; специалист-аналитик – штатный сотрудник, дипломник.

3.2 Планируемая себестоимость разработки информационной системы

Плановая совокупная стоимость разработки и внедрения ИС определяются по следующим основным статьям:

1. Заработная плата сотрудников, специалистов, задействованных в проекте.

2. Страховые взносы.
3. Затраты организации на вычислительную технику
4. Затраты организации на расходные материалы.
5. Расходы организации на эксплуатацию технических средств.

Прогнозируемые затраты на оплату труда вычисляются на основании продолжительности этапов создания ИС и стоимости одного рабочего дня участника проекта, вычисляемого на основании его оклада. Поскольку работы по созданию ИС не являются основной деятельностью каждого из участников проекта, их стоимость оценивается в размере $\frac{1}{2}$ ставки рабочего дня.

Информация о заработной плате сотрудников, задействованных в проекте, а также стоимость их одного рабочего дня, приведены в таблице 17.

Таблица 17. Планируемая сумма оплаты труда специалистов

Должность сотрудника	Оклад сотрудника, руб.	Оплата сотрудника в разрезе за 1 рабочий день (0,5 ставки), руб.
Руководитель проекта	50 000,00	1 562,5
Программист	35 000,00	514,70
Специалист-аналитик	40 000,00	571,42

Плановые суммарные затраты на оплату труда исчисляются на основании данных из таблиц 16 и 17, путем перемножения количества дней, когда каждый из сотрудников был занят проектом на стоимость одного рабочего дня. Полученные расчеты представлены в таблице 18.

Таблица 18. Планируемые суммарные затраты на оплату труда

Должность сотрудника	Оплата за 1 рабочий день (0,5 ставки), руб.	Кол-во рабочих дней	Затраты на оплату труда, руб.
Руководитель проекта	1 562,5	28	43750,00
Программист	514,70	50	25735,00
Специалист-аналитик	571,42	52	29713,84
Итого:			99198,84

Тарифы страховых взносов в обязательные фонды в 2012-2016 годах для основной группы налогоплательщиков в процентах от налогооблагаемой базы составляют:

1. Пенсионный фонд Российской Федерации - 22%.
2. Фонд социального страхования Российской Федерации - 2,9%.
3. Федеральный фонд обязательного медицинского страхования - 5,1%.

По итогам, сумма всех страховых взносов составила 30 %.

Согласно данным, с учётом планируемых суммарных затрат на оплату труда специалистов можем рассчитать сумму отчислений:

$$O = S \cdot r = 99198,84 \cdot 0,30 = 29759,65 \text{ руб.},$$

где, O - итоговая сумма страховых отчислений;

S - совокупная сумма оплаты труда сотрудников, задействованных в проекте;

r - совокупная ставка страховых взносов.

В процессе разработки ИС используется один персональный компьютер, стоимость и характеристики которого указаны в таблице 19.

Таблица 19. Стоимость технических средств разработки ЭИС

Наименование технического средства	Стоимость, руб.
Системный блок: Intel Pentium, CPU 3.0 ГГц, HDD: 500 Гб, ОЗУ DDR3 4096 Мб 1600 МГц, видео 1024 Мб	30110
Клавиатура A4Tech USB	500
Монитор LCD BenQ GL955A	7000
Мышь optical A4Tech USB	450
Принтер Canon LBP 6000	4800
Итого:	42860

Следующей статьей для формирования планируемой себестоимости разработки ИС являются затраты на расходные материалы. В таблице 20 представлена структура затрат на канцелярские товары и картридж.

Таблица 20. Затраты на канцтовары и картридж

№ п/п	Наименование материала	Ед. изм.	Количество, шт.	Цена, руб.	Общая стоимость, руб.
1	Бумага форм. А4	уп.	4	250	1000
2	Картридж лазерного принтера СВ435А	шт.	1	2000	2000
3	Лампа настольная	шт.	4	700	2100
4	Ручка шариковая	шт.	10	30	300
6	Коврик для мыши	шт.	5	200	1000
8	Скрепки маленькие	уп.	2	60	120
9	Бумажные закладки	уп.	3	200	600
10	Степлер средний	шт.	3	250	750
11	Скотч большой	шт.	2	100	200
12	Органайзер	шт.	5	200	1000
14	Карандаши	шт.	15	10	150
16	Итого				9220

Далее рассмотрим и оценим статью эксплуатационных расходов. Эксплуатационные расходы - издержки, связанные с поддержанием работоспособного состояния используемых в ходе разработки систем и прочего оборудования в течение всего намеченного срока службы. К эксплуатационным расходам при проектировании системы можно отнести затраты на оплату электроэнергии при пользовании вычислительной техникой сотрудниками, вовлечёнными в проект.

В таблице 21 приведен расчет планируемых затрат на электроэнергию учетом мощности блоков питания персональных компьютеров и времени их работы.

Таблица 21. Расчет планируемых затрат на электроэнергию

Модель (тип) блока питания	Потребляемая мощность, кВт	Количество часов работы	Количество кВт	Стоимость кВт/ч, руб.	Сумма, руб.
FSP 550PNR	0,55	360	250	5,03	663,96
Итого:					663,96

Планируемая себестоимость разработки ИС складывается из рассмотренных выше статей затрат. В таблице 22 представлена структура затрат на разработку и их доля от общей стоимости системы.

Таблица 22. Структура плановой себестоимости ЭИС

Статьи затрат	Сумма, руб.	Доля от общей суммы затрат
Заработная плата разработчиков	99198,84	54,59
Страховые взносы	29759,65	16,38
Затраты на вычислительную технику	42860	23,59
Затраты на расходные материалы	9220	5,07
Эксплуатационные расходы	663,96	0,37
Итого:	181702,5 руб.	

Таким образом, стоимость системы составит примерно 182 000 руб.

3.3. Обоснование экономической целесообразности разработки и внедрения информационной системы

Расчет себестоимости, разрабатываемой ИС приема и анализа заявок отделом техподдержки ЧП «Net Soft System» показал, что стоимость системы составит примерно 182 тыс. руб. В данную стоимость входят все затраты на внедрение, сопровождение и доработку системы в случае обнаружения сбоев и ошибок.

Для оценки экономической целесообразности разработки собственной системы, необходимо оценить стоимость ее аналогов в год в расчете на 100 пользователей (таблица 23).

Таблица 23. Сравнение программных продуктов по их стоимости

Система	Оценка стоимости	Стоимость, руб./год
«WorkFlowSoft»	1 лицензия – 350 руб.	350 000
«1С» - «ServiseDesk»	от 20000 до 40000 руб. в месяц	360 000
«ServiceNow»	20000 руб. в месяц	240 000
«HelpDesk»	от 10 000 до 25 000 руб. в месяц	210 000
Собственная разработка	-	182 000

Согласно данным в таблице 23, стоимость всех рассматриваемых аналогов, выше стоимости разрабатываемой ЭИС.

В отличие от готовых разработок, собственное решение полностью удовлетворяет всем заявленным требованиям ЧП «Net Soft System».

Еще одним не маловажным преимуществом собственной разработанной системы является:

- обеспечение безопасности данных, которые будут храниться непосредственно на сервере компании;
- возможность в любой момент модифицировать программный продукт;
- независимость от внешних каналов связи;
- собственные наработки, защита авторских прав;

Выводы по главе 3

В рамках экономического раздела:

- описаны основные этапы разработки системы;
- выявлен состав участников проекта: руководитель, программист и специалист-аналитик;
- определен срок разработки системы: 90 дней и составлен график выполнения работ;
- рассчитана себестоимость системы: 182000 руб.;
- произведена оценка экономической целесообразности разработки собственной системы.
- выявлены положительные стороны собственного продукта

ЗАКЛЮЧЕНИЕ

Для дальнейшего, успешного и прибыльного функционирования ЧП «Net Soft System» в условиях растущей конкуренции и требований со стороны клиентов к уровню и качеству оказываемых услуг, необходимо внедрение системы для информационно-технической поддержки пользователей.

Существующие на отечественном рынке аналогичные решения не соответствуют требованиям, предъявленным исследуемой компанией, в связи с чем, было принято решение о разработке новой и качественной системы, ориентированной конкретно под бизнес-процессы ЧП «Net Soft System».

Проведенный анализ предметной области позволил выявить процессы учета, требующие автоматизации: создание, регистрация и исполнение заявок, контроль их сроков исполнения, учет пользователей, формирование отчетов и т.п.

Анализ организационной структуры компании позволил выявить роли пользователей в разрабатываемой системе (пользователь, ответственный и исполнитель) и обозначить задачи, которые выбранные группы пользователей смогут решать в рамках ИС.

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

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

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

Для разработки системы была выбрана среда программирования «Embarcadero RAD Studio XE10 Architect», так как данный язык является

основным и используется штатными программистами и разработчиками на предприятии, а также использована СУБД «MySQL», так как компания уже много лет широко использует его.

Разработанная система состоит из 41 формы. База данных включает 11 таблиц. Программа делится на три модуля: «Пользователь», «Ответственный» и «Исполнитель». Доступ к каждому модулю осуществляется посредством ввода логина и пароля.

ИС обеспечивает решение следующих задач: учет клиентов и пользователей; создание заявок и отправка их на регистрацию; назначение/изменение исполнителя заявки; регистрация заявок с присвоением номера; обработка и исполнение заявок; назначение сроков исполнения заявок согласно приоритетам; настройка и продление сроков исполнения заявок; обмен сообщениями; обмен графическими материалами; автоматическая смена статуса заявки; уведомление о новых сообщениях и заявках; формирование расписания задач и контроль их исполнения; учет сотрудников; формирование отчетов и т.д.

Себестоимость разработанной системы составила 182 000 руб., что доказывает целесообразность ее разработки собственными силами.

Для облегчения работы пользователя с программой, предусмотрено «Руководство пользователя», поэтапно описывающее и наглядно демонстрирующее ход выполнения каждой операции в программе. Простой и удобный интерфейс программы позволяет разобраться в ней любому пользователю, не требуя специальной компьютерной подготовки.

Все кнопки программы сопровождаются всплывающими подсказками, что придает пользователю большую уверенность при работе с системой на начальных этапах ее внедрения.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. ГОСТ 34.601-90 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания»
2. ГОСТ 34.602-89 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»
3. Агальцов, В.П. Базы данных. В 2-х т. Т. 2. Распределенные и удаленные базы данных: Учебник / В.П. Агальцов. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 272 с.
4. Вдовин В. М. Предметно-ориентированные экономические информационные системы [Электронный ресурс]: учебное пособие / В. М. Вдовин, Л. Е. Суркова, А. А. Шурупов. - 3-е изд. - Москва: Дашков и К°, 2013. - 388 с.: ил. - ISBN 978-5-394-02262-3.
5. Крышкин О. Настольная книга по внутреннему аудиту: Риски и бизнес-процессы / О. Крышкин. – М.: АЛЬПИНА ПАБЛИШЕР, 2013. – 477 с.
6. Избачков Ю.С. Информационные системы: учебник для вузов / Ю.С. Избачков, В.Н. Петров. - 2-е изд.- Санкт-Петербург: Питер, 2015. – 656 с.
7. Карпова, И.П. Базы данных: Учебное пособие / И.П. Карпова. - СПб.: Питер, 2013. - 240 с.
8. Нив Г. Организация как система: Принципы построения устойчивого бизнеса Эдвардса Деминга: пер. с англ. / Г. Нив. – 3-е изд. – М.: АЛЬПИНА ПАБЛИШЕР, 2014. – 368 с.
9. Ларман К. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку. – М.: Вильямс, 2013. – 736с.
10. Пирогов, В.Ю. Информационные системы и базы данных: организация и проектирование: Учебное пособие / В.Ю. Пирогов. - СПб.: БХВ-Петербург, 2009. - 528 с.

11. Шерр Август-Вильгельм. Моделирование бизнес - процессов. М.: Весть-МетаТехнология, 2010. -175 С.
12. ЧП «Net Soft System» [Электронный ресурс]. URL: company/net-soft-system (дата обращения: 05.03.2018).
13. ЧП «Net Soft System». Устав компании.
14. Программное обеспечение Embarcadero Rad Studio [Электронный ресурс]: URL: <https://www.embarcadero.com/ru/products/rad-studio>.
15. Система «1С: ServiceDesk» [Электронный ресурс]. URL: <http://www.audit-escort.ru/programms/servicedesk-sluzhba-podderzhki/#tabs1-css> (дата обращения: 25.11.2017).
16. Columbus ERP Market Share Update Forbes [Электронный ресурс] : Режим доступа : <http://www.forbes.com/sites/louiscolombus/2013/04/26/2013-erp-market-share-update-40-of-crm-systems-sold-are-saas-based>.
17. Foss B. Journal of Database Marketing & Customer Strategy Management / B. Foss, M. Stone, Y. Ekinici. – New York City: ACM, 2013. – P. 315.
18. George V. BPM: Good or Bad for IT. [Электронный ресурс] / V. George. – Режим доступа: <http://www.amazon.com>.
19. Hofmann T. Latent semantic models for collaborative filtering / T. Hofmann. – New York City : ACM Trans. – 2014. – P. 115.
20. Hommes B. The Evaluation of Business Process Modeling Techniques / B. Hommes. –New York City: ACM, 2012. – P. 20.

ПРИЛОЖЕНИЯ

Приложение 1

Описание обозначений базы данных

Primary key (PRI) – первичный ключ.

Foreign key (MUL) – внешний ключ.

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

Auto_increment – данное свойство обеспечивает автоматическую нумерацию строк таблицы.

Описание типов данных:

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

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

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

Mediumblob – строковый тип. Текст длиной от 0 до 16777215 0символов. Используется для графических материалов.

Фрагменты листинга программы

```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
  Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, XPMani;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    BitBtn2: TBitBtn;
    Edit5: TEdit;
    Edit6: TEdit;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Edit3: TEdit;
    Edit4: TEdit;
    Button4: TButton;
    Button5: TButton;
    XPManiFest1: TXPManifest;
    procedure BitBtn2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

uses Unit21, Unit3, Unit4, Unit32, Unit11, Unit16,
  Unit15, Unit34, Unit30;

{$R *.dfm}

procedure TForm1.BitBtn2Click(Sender:
TObject);
begin
  if (Edit1.Text=EmptyStr) or
  (Edit2.Text=EmptyStr) then begin
    ShowMessage(Введите логин и пароль);
  end
  else begin
    dm.t7.Close;
    dm.t7.SQL.Clear;
    dm.t7.SQL.Add(Select * from sotrudniki where
    login LIKE '"+Edit1.Text+"' and parol LIKE
    '"+Edit2.Text+"'");
    dm.t7.Open;
    if dm.t7.RecordCount=0 then begin
      ShowMessage(Неверный логин или пароль.
      Попробуйте еще раз.);
    end
    else begin
      Edit3.Text:=dm.t7.FieldByName(id_sotrud).Value;
      Edit5.Text:=dm.t7.FieldByName(fio).Value;
      Edit6.Text:=dm.t7.FieldByName(rol).Value;
      Edit4.Text:=dm.t7.FieldByName(id_otdela).Value;
      Button1.Click;
    end
  end
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  if Edit6.Text=Исполнитель' then begin
    Button2.Click;
  end
  else if Edit6.Text=Пользователь' then begin
    Button3.Click;
  end
  else if Edit6.Text=Ответственный' then begin
    Button4.Click;
  end
  else begin Form16.Show;
  end
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  Form3.WindowState:=wsMaximized;
  Form3.Label1.Caption:=Edit5.Text;
  Form3.Label2.Caption:=Edit3.Text;
  Form3.Label3.Caption:=Edit4.Text;
  Form32.Edit1.Text:=Edit3.Text;
  Form3.show;
  Form34.ToolButton4.Click;
  Form1.Hide;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  Form4.WindowState:=wsMaximized;
  Form4.Label1.Caption:=Edit5.Text;
  Form4.Label2.Caption:=Edit3.Text;
  Form4.Label3.Caption:=Edit4.Text;
  Form32.Edit1.Text:=Edit3.Text;
  Form4.show;
  Form30.ToolButton4.Click;
  Form1.Hide;
end;
end;

```

```

procedure TForm1.Button4Click(Sender: TObject);
begin
Form11.WindowState:=wsMaximized;
Form11.Label1.Caption:=Edit5.Text;
Form11.Label2.Caption:=Edit3.Text;
Form11.Label3.Caption:=Edit4.Text;
Form32.Edit1.Text:=Edit3.Text;
Form11.show;
Form15.ToolButton4.Click;
Form1.Hide;
end;

end.

unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
  Graphics, Controls, Forms,
  Dialogs, jpeg, ExtCtrls;

type
  TForm2 = class(TForm)
    Timer1: TTimer;
    Image1: TImage;
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

{$R *.dfm}

procedure TForm2.Timer1Timer(Sender: TObject);
begin
Timer1.Enabled := false;
end;

end.

unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
  Graphics, Controls, Forms,
  Dialogs, Menus, StdCtrls, ExtCtrls;

type
  TForm3 = class(TForm)
    N15Click(Sender: TObject);
  begin
Form17.Parent:=Form3.ScrollBox1;
Form17.show;
end;

```

```

TForm3 = class(TForm)
  ScrollBox1: TScrollBox;
  Panel1: TPanel;
  Label1: TLabel;
  Label2: TLabel;
  MainMenu: TMainMenu;
  N3: TMenuItem;
  N9: TMenuItem;
  N10: TMenuItem;
  N21: TMenuItem;
  N22: TMenuItem;
  Label3: TLabel;
  N1: TMenuItem;
  procedure N11Click(Sender: TObject);
  procedure N26Click(Sender: TObject);
  procedure N7Click(Sender: TObject);
  procedure N15Click(Sender: TObject);
  procedure N8Click(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action:
  TCloseAction);
  procedure N22Click(Sender: TObject);
  procedure N10Click(Sender: TObject);
  procedure N9Click(Sender: TObject);
  procedure N18Click(Sender: TObject);
  procedure N1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

uses Unit21, Unit5, Unit7, Unit9, Unit11, Unit13,
Unit15, Unit17, Unit19,
Unit22, Unit25, Unit1, Unit30, Unit34, Unit36, Unit35,
Unit37, Unit38,
Unit39, Unit31, Unit40;

{$R *.dfm}

procedure TForm3.N11Click(Sender: TObject);
begin
Form11.Parent:=Form3.ScrollBox1;
Form11.show;
end;

procedure TForm3.N26Click(Sender: TObject);
begin
Form13.Parent:=Form3.ScrollBox1;
Form13.show;
end;

procedure TForm3.N7Click(Sender: TObject);
begin
Form15.Parent:=Form3.ScrollBox1;
Form15.show;
end;

```

```

procedure TForm3.N8Click(Sender: TObject);
begin
  rm25.Parent:=Form3.ScrollBox1;
  rm25.show;
end;

procedure TForm3.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  rm1.close;
end;

procedure TForm3.N22Click(Sender: TObject);
begin
  rm34.Parent:=Form3.ScrollBox1;
  rt5.Close;
  rt5.SQL.Clear;
  rt5.SQL.Add('SELECT * FROM zayavki where
  _ispolnit like "'+Form3.Label2.Caption+'" AND (status
  = "Зарегистрована")');
  rt5.Open;
  rm34.Show;
  rm31.ToolButton10.Visible:=false;
  rm31.ToolButton11.Visible:=false;
  rm31.DEEdit6.Enabled:=false;
  rm31.BitBtn1.Visible:=false;
  rm31.BitBtn3.Visible:=false;
  rm31.BitBtn5.Visible:=false;
  rm31.DELookupComboBox2.Enabled:=false;
  rm31.DELookupComboBox1.Enabled:=false;
  rm31.DEEdit8.Enabled:=false;
  rm31.DEMemo1.Enabled:=false;
  rm31.cxGridDBTableView3.Column2.Visible:=false;
end;

procedure TForm3.N10Click(Sender: TObject);
begin
  rm36.Parent:=Form3.ScrollBox1;
  rm36.WebBrowser1.Navigate(ExtractFilePath(Application
  ExeName) + 'Помощь.htm');
  rm36.Show;
end;

procedure TForm3.N9Click(Sender: TObject);
begin
  rm35.Parent:=Form3.ScrollBox1;
  rm35.Show;
end;

procedure TForm3.N18Click(Sender: TObject);
begin
  rm22.Parent:=Form3.ScrollBox1;
  rm22.show;
end;

procedure TForm4.N9Click(Sender: TObject);
begin
  rm35.Parent:=Form4.ScrollBox1;
  rm35.Show;
end;

procedure TForm3.N1Click(Sender: TObject);
begin
  Form40.Parent:=Form3.ScrollBox1;
  dm.tl9.Close;
  dm.tl9.SQL.Clear;
  dm.tl9.SQL.Add('Select * from sotrudniki where
  id_sotrud LIKE "'+Label2.Caption+'"');
  dm.tl9.Open;
  Form40.show;
end;

end.

unit Unit4;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
  Graphics, Controls, Forms,
  Dialogs, Menus, StdCtrls, ExtCtrls;

type
  TForm4 = class(TForm)
    ScrollBox1: TScrollBox;
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    MainMenu1: TMainMenu;
    N3: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    Label3: TLabel;
    N4: TMenuItem;
    procedure FormClose(Sender: TObject; var Action:
    TCloseAction);
    procedure N9Click(Sender: TObject);
    procedure N10Click(Sender: TObject);
    procedure N6Click(Sender: TObject);
    procedure N4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form4: TForm4;

implementation

uses Unit21, Unit1, Unit30, Unit35, Unit36, Unit26,
  Unit40, Unit3, Unit31;

{$R *.dfm}

procedure TForm4.FormClose(Sender: TObject; var
Action: TCloseAction);
begin
  Form1.close;

```