

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение
высшего образования

**«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»**

Факультет Электротехнический

Кафедра автоматики

Направление подготовки 27.03.04 «Управление в технических системах»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой

Голощапов С. С.

2021 г.

Программное обеспечение системы обработки претензий и рекламаций
на производственном предприятии

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ– 27.03.04.2021.799.00.00ПЗ ВКР

Автор ВКР

студент группы МиЭт - 529

подпись

/ Гоголев А.О.

ФИО

2021 г.

Руководитель работы

д. физ.-мат.н., профессор

должность

подпись

/ Телегин А.И.

ФИО

2021 г.

Консультант

Нач. отдела автоматизации ООО ПФ «Ника»

должность

подпись

/ Дресвянников М.В.

ФИО

2021 г.

Нормоконтроль

старший преподаватель

должность

подпись ФИО

/ Елисеев В.П.

2021 г.

АННОТАЦИЯ

Гоголев А.О. Выпускная квалификационная работа «Программное обеспечение системы обработки претензий и рекламаций на производственном предприятии» - Миасс: ЮУрГУ - 2021. - 85 с, 16 илл., 7 табл., библиогр. список - 16 наим.

В ходе написания работы разработано программное обеспечение системы обработки претензий и рекламаций для производственного предприятия, разработаны структура базы данных и интерфейс доступа конечного пользователя. Программное обеспечение реализовано на языке C++ и представляет собой исполняемый файл для систем Microsoft Windows. Приложение автоматизирует процесс доступа и управления информацией в базе данных с заданной структурой, обеспечивает контроль наличия задач, требуемых выполнения, сроков выполнения задач с выводом уведомлений на экран.

					27.03.04.2021.799.00.00 ПЗ ВКР			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Гоголев А.О.			Программное обеспечение системы обработки претензий и рекламаций на производственном предприятии	Лит.	Лист	Листов
Провер.		Дресвянников М.В.				К	5	85
Н. Контр.		Елисеев В.П.				ЮУрГУ		
Утверд.		Голощапов С.С.				Кафедра «Автоматика»		

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	8
1 ОПРЕДЕЛЕНИЕ И КЛАССИФИКАЦИЯ СИСТЕМ АВТОМАТИЗАЦИИ	10
2 ОБЗОР СУЩЕСТВУЮЩИХ АНАЛОГОВ СИСТЕМ ДОКУМЕНТООБОРОТА	13
2.1 Общая классификация систем документооборота	15
2.2 Обзор основных систем документооборота, представленных в России.....	17
2.2.1 OpenDocMan	20
2.2.2 Document Pro.....	21
2.2.3 Другие системы	23
2.3 Проблемы внедрения системы документооборота.....	25
3 ОБОСНОВАНИЕ НЕОБХОДИМОСТИ СОЗДАНИЯ НОВОГО ПО ДЛ	
ПРЕДПРИЯТИЯ И ПОСТАНОВКА ЗАДАЧИ	27
3.1 Термины, определения согласно стандарту, устанавливаемому организацию и порядок работы с претензиями потребителей	27
3.2 Общие данные	27
3.3 Сроки работы с претензиями	31
3.4 Постановка задачи.....	32
4 ТИПЫ И СРАВНЕНИЕ МОДЕЛЕЙ БАЗ ДАННЫХ	33
4.1 Иерархическая модель.....	33
4.2 Сетевая модель	33
4.3 Реляционная модель.....	34
4.4 Сравнение трёх моделей.....	36
4.5 Другие модели баз данных (ООСУБД).....	36
5 СРАВНЕНИЕ И ВЫБОР СУБД.....	39
5.1 MySQL.....	39
5.2 PostgreSQL	39
5.3 SQLite	41
5.4 Выбор конкретной СУБД.....	42
6 РЕАЛИЗАЦИЯ МОДЕЛИ БД, ОПИСАНИЕ РАБОТЫ ПРОГРАММЫ, РАЗРАБОТКА ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ.....	43

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

6.1 Проектирование структуры базы данных.....	43
6.2 Реализация модели БД.....	44
6.3 Разработка интерфейса пользователя	52
6.3.1 Создание окна входа в базу данных	53
6.3.2 Создание основного окна базы данных	59
6.3.3 Создание окна «История активности пользователей» базы данных	70
6.3.4 Создание окна «Сопроводительные документы» базы данных	73
6.3.5 Создание окна изменения документа в базе данных.....	76
6.4 Описание программы.....	79
6.5 Поставка созданного приложения.....	79
6.6 Перенос созданного приложения в архитектуру клиент-сервер.....	81
ЗАКЛЮЧЕНИЕ	83
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	84

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

ВВЕДЕНИЕ

Рост сектора информации является лишь одной причиной, по которой управление процессом связи приобретает для производительности все более важный характер. Технический прогресс в обработке информации – компьютеры, спутники связи, всемирная телекоммуникационная сеть – радикальнейшим образом увеличили объемы обрабатываемой информации и сократили время на ее передачу.

Воздействие роста объема информации и сокращение времени ее передачи не совсем однозначно. Ясно, что улучшение качества информации, имеющейся в момент принятия решения, позволяет руководству принять обоснованное, своевременное решение. Немедленная передача подробной информации способствует координации деятельности физически разобщенных подразделений. Однако огромный объем циркулирующей в настоящее время информации все больше затрудняет нахождение и выделение нужных и относящихся к делу сведений. Сокращение времени передачи информации означает, что у менеджеров остается все меньше времени на ее получение и использование. Информация является одним из основных ресурсов роста производительности. Более эффективное использование информации приобретает все более важное значение для обеспечения производительности организации в целом.

Информационные технологии – это совокупность методов, производственных процессов и программно-технических средств, объединенных в технологическую цепочку, обеспечивающую сбор, обработку, хранение, распространение (транспортировку) и отображение информации с целью снижения трудоемкости процессов использования информационного ресурса, а также повышения их надежности и оперативности.

Процесс автоматизации представляет собой совокупность методических, языковых, технических и программных средств, позволяющих организовать работу конечных пользователей в некоторой предметной области.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

Основные преимущества автоматизации и новой технологии переработки информации сказываются там, где приходится выполнять повторяющиеся задачи, предусматривающие запрограммированные решения, либо задачи с большим объемом вычислений или чисто механического труда. Такие задачи составляют достаточно большую часть работы. Компьютерная техника позволяет ускорить почти любой процесс.

Процесс автоматизации деятельности отдела предприятия включает такие этапы как разработка технического задания, описание и проектирование процессов, создание базы данных, кодирование пользовательского интерфейса, тестирование и отладка системы автоматизации. Дальнейшее развитие системы автоматизации предусматривает анализ и сопровождение системы.

У каждого предприятия своё направление работы, специфика, оборудование. Программное обеспечение автоматизированной системы управления на предприятии разрабатывается в зависимости от поставленных задач. Существуют стандартные пакеты, которые необходимо только настроить на конкретное предприятие, но также существует и необходимость создания собственного программного обеспечения в рамках реализуемой задачи.

Актуальность работы определяется необходимостью разработки программного обеспечения, реализуемого в режиме диалога с использованием средств программирования, для повышения качества и скорости обработки информации по претензиям и рекламациям на предприятии ООО ПФ НИКА.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

1 ОПРЕДЕЛЕНИЕ И КЛАССИФИКАЦИЯ СИСТЕМ АВТОМАТИЗАЦИИ

Последние тенденции в промышленных и домашних системах управления свидетельствуют о развитии технологий автоматизации. Несмотря на скудное начало в последние десятилетия, автоматизация перешла от проводного к беспроводному управлению над простыми и сложными приложениями управления. С появлением различных типов систем автоматизации характер управления технологическим процессом становится все более прогрессивным, чтобы обеспечить более высокую производительность при более низких затратах.

Автоматизация относится к состоянию автоматического управления оборудованием или машинами и представляет собой шаг за пределы механизации, когда требуется физический труд для выполнения функций вручную. Благодаря автоматизации оборудования ручной труд сокращается за счет использования компьютерного и автоматизированного оборудования. Работа системы автоматизации интегрирована с использованием сенсорных систем, контуров управления с обратной связью и автоматических исполнительных устройств.

«Автоматизация лишает людей работы» - распространенное заблуждение об автоматизации. Но причина автоматизации технологического процесса заключается в повышении производительности и качества продукции, сокращении количества ручных / периодических проверок, повышении безопасности, снижении производственных затрат и удобстве для оператора. Автоматизация предоставляет мощные решения для управления широким спектром приложений, таких как промышленность, автомобилестроение, авиация, бытовая техника и т. Д.

В автоматическом управлении для различных операций используются современные контроллеры, такие как программируемые логические контроллеры (ПЛК). Эти контроллеры запрограммированы с усовершенствованными контурами управления, которые являются его неотъемлемой частью, и некоторые из этих структур управления представляют собой обратную связь, прямую связь, каскад, контуры управления соотношением и так далее.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

2 типа систем автоматизации Система автоматизации заменяет обычную проводную релейную систему, автоматизируя функции процесса с использованием различных аппаратных и программных средств с минимальным вмешательством человека. В разных приложениях используются разные типы систем автоматизации, но все автоматизированные системы не похожи. В зависимости от области применения и характера управления эти автоматизированные системы классифицируются:

1. Промышленная автоматизация - это процесс повышения гибкости, упрощения и повышения эффективности процессов промышленного производства.

Интеграция автоматизации в отрасли приводит к появлению интеллектуальных производственных решений с улучшенным качеством продукции и производительностью, а также сокращением времени простоя и отходов. Промышленная автоматизация предполагает использование различных устройств управления, таких как ПК / ПЛК, различных датчиков и исполнительных механизмов, коммуникационных шин / модулей, приводов машин, систем НМІ (человеко-машинного интерфейса) и другого управляющего оборудования.

Индустриальная автоматизация - этот тип автоматизации наиболее широко используется в автомобильной, компьютерной и электронной, медицинской, телекоммуникационной, потребительской и других отраслях промышленности.

Системы автоматизации могут быть фиксированными, программируемыми, гибкими и интегрированными. Ниже приведены некоторые виды промышленной автоматизации.

- Машины с ЧПУ. Эти машины представляют собой машины с компьютерным управлением, которые используют компьютеры для выполнения операций управления путем сбора, обработки, вычисления и управления переменными процесса. Автоматизация представляет собой программную версию станков, также называемых станками с числовым программным управлением (ЧПУ). Станки с ЧПУ используются для резки и фрезерования для обеспечения высокой точности и точной прецизионной работы.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

- Компьютерное производство (САМ). При этом весь производственный процесс (включая производство, планирование и контроль) автоматизирован с использованием машин с числовым программным управлением, промышленных роботов и других типов устройств автоматизации. Эти системы автоматизации также используют компьютеры для планирования, проектирования и компоновки различных продуктов. Примерами таких систем автоматизации являются автоматизированное проектирование (САД), автоматизированное проектирование и черчение (САДД) и автоматизированное планирование процессов (САРР).

- Промышленные роботы. Тип автоматизированных машин или оборудования, которые могут выполнять различные задачи в течение более длительного времени. Чаще всего они реализуются в очень опасных или опасных для человека областях.

- Гибкие производственные системы. Начиная с процесса планирования и проектирования и заканчивая отгрузкой продукции, вся система полностью интегрирована для автоматизации. Автоматизация объединяет машины с числовым программным управлением, промышленных роботов и другое оборудование автоматизации в одну интегрированную систему.

2. Домашняя автоматизация. Поскольку технологии быстро развиваются, это влияет на повседневную жизнь человека, особенно в управлении бытовой техникой. Система домашней автоматизации выполняет такие операции, как управление и регулирование освещения, регулирование оборудования HVAC (отопления, вентиляции и кондиционирования воздуха), управление энергопотреблением и нагрузкой, безопасность и мониторинг, аудио / видео системы, НМІ и другие задачи управления.

Это два типа систем автоматизации, которые внедряются в большинстве отраслей промышленности и дома. Общее качество, точность, производительность, снижение затрат на рабочую силу и безопасность - лучшие аргументы в пользу систем автоматизации.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

2 ОБЗОР СУЩЕСТВУЮЩИХ АНАЛОГОВ СИСТЕМ ДОКУМЕНТООБОРОТА

Аналогично тому, как бит является единицей информации в кибернетике, документ является единицей информации в системах документооборота. Системы документооборота хранят документы, ведут их историю, обеспечивают их движение по организации, позволяют отслеживать выполнение тех бизнес-процессов, к которым эти документы имеют отношение. В организации, где внедрена система документооборота, документ является базовым инструментом управления. Здесь нет просто решений, поручений или приказов - есть документы, содержащие эти самые приказы, решения, поручения и т. д.: все управление в организации осуществляется через документы.

Любой документ в системе документооборота снабжается "карточкой", подобной библиотечной. Обычно конкретный набор полей в такой карточке привязан к типу документа. Хранилище системы электронного документооборота можно представить себе в виде базы данных, содержащей информацию полей карточек и некоторого хранилища для самих документов.

Системы документооборота обычно внедряются, чтобы решать определенные задачи, стоящие перед организацией, из которых наиболее часто встречаются следующие:

- Обеспечение более эффективного управления за счет автоматического контроля выполнения, прозрачности деятельности всей организации на всех уровнях.
- Поддержка системы контроля качества, соответствующей международным нормам.
- Поддержка эффективного накопления, управления и доступа к информации и знаниям. Обеспечение кадровой гибкости за счет большей формализации деятельности каждого сотрудника и возможности хранения всей предыстории его деятельности.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

- Протоколирование деятельности предприятия в целом (внутренние служебные расследования, анализ деятельности подразделений, выявление "горячих точек" в деятельности).

- Оптимизация бизнес-процессов и автоматизация механизма их выполнения и контроля.

- Исключение или максимально возможное сокращение оборота бумажных документов на предприятии. Экономия ресурсов за счет сокращения издержек на управление потоками документов в организации.

- Исключение необходимости или существенное упрощение и удешевление хранения бумажных документов за счет наличия оперативного электронного архива.

Основные факторы, влияющие на решение о выборе системы:

- Требования по объему хранения. Если у вас много документов (по объему хранения), необходимо выбрать систему, поддерживающую иерархическое структурное хранение (HSM - Hierarchal Storage Management). Этот механизм хранит наиболее активно используемые данные на наиболее быстрых, но и наиболее дорогих носителях, в то время как реже используемая информация автоматически переносится на медленные и дешевые носители.

- Наличие формализуемых процедур, требующих поддержки их выполнения и автоматизации контроля (подготовки документов определенного типа, выполнения стандартных функций организации и т. д.).

- Необходимость автоматизации административного управления организацией. Степень сложности организационной структуры.

- Наличие территориально распределенных подразделений. Этот фактор накладывает определенные требования к удаленному доступу, к репликации данных и т.д.

- Наличие бумажного архива большого объема. Некоторые системы документооборота поставляются с уже интегрированными подсистемами массового ввода документов.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

- Наличие не удовлетворяющей текущим потребностям системы документооборота.

- Необходимость в развитой маршрутизации документов, в управлении потоками работ (workflow managing). Как продолжение этой необходимости - потребность в поддержке произвольных бизнес-процессов, возможно работающих совместно с прикладными системами поддержки этих процессов.

- Требования по срокам хранения документов. При больших сроках хранения (десятки лет) стоит серьезно подумать об организации параллельного архива на микрофильмах.

- Требования к "открытости", расширяемости системы. Возможность интеграции с существующими информационными системами и использования имеющегося оборудования.

- Необходимость хранения изображений документов. Использование в организации специфических форматов хранения документов. Необходимость поддержки инженерных и конструкторских задач, других особенностей деятельности предприятия.

- Необходимость развитых средств поиска информации. Полная поддержка системой языков имеющихся в организации документов.

- Требования к безопасности (шифрование, организация доступа и т. д.). Возможность использования уже имеющихся в информационной инфраструктуре организации механизмов доступа в системе документооборота.

- Требования по соответствию определенным стандартам: внутренним, отраслевым, ГОСТ, международным стандартам по контролю качества, уровню организации хранения информации.

2.1 Общая классификация систем документооборота

Любая система документооборота может содержать элементы каждой из приведенных ниже категорий, но большинство из них имеют конкретную

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

ориентацию в одной из областей, связанную в первую очередь с позиционированием продукта.

- Системы с развитыми средствами хранения и поиска информации (электронные архивы - ЭА). Электронный архив - это частный случай системы документооборота, ориентированный на эффективное хранение и поиск информации. Некоторые системы особенно выделяются за счет развитых средств полнотекстового поиска: нечеткий поиск, смысловой поиск и т. д., другие - за счет эффективной организации хранения: HSM, поддержка широкого диапазона оборудования для хранения информации и т. д.

- Системы с развитыми средствами workflow (WF). Эти системы в основном рассчитаны на обеспечение движения неких объектов по заранее заданным маршрутам (так называемая "жесткая маршрутизация"). На каждом этапе объект может меняться, поэтому его называют общим словом "работа" (work). Системы такого типа называют системами workflow - "поток работ". К работам могут быть привязаны документы, но не документы являются базовым объектом этих систем. С помощью таких систем можно организовать определенные работы, для которых заранее известны и могут быть прописаны все этапы.

- Системы, ориентированные на поддержку управления организацией и накопление знаний. Эти "гибридные" системы, которые обычно сочетают в себе элементы двух предыдущих. При этом базовым понятием в системе может быть, как сам документ, так и задание, которое нужно выполнить. Для управления организацией нужна как "жесткая", так и "свободная" маршрутизация, когда маршрут движения документа назначает руководитель, поэтому обе технологии в том или ином виде могут присутствовать в таких системах. Эти системы активно используются в государственных структурах управления, в офисах крупных компаний, которые отличаются развитой иерархией, имеют определенные правила и процедуры движения документов. При этом сотрудники коллективно создают документы, готовят и принимают решения, исполняют или контролируют их исполнение. При внедрении таких систем на крупных предприятиях важно определить, предоставляет ли система возможность эффективного

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		16

администрирования, обработки больших объемов информации, интеграции с автоматизированными системами управления производством, масштабируемости, поэтапного внедрения, учета территориальной распределенности, сложной организационной структуры, ролевого принципа организации доступа и т. д.

- Системы, ориентированные на поддержку совместной работы (collaboration). Это новое веяние в области систем документооборота, связанное с пониманием изменчивости рыночных условий в современном мире и с необходимостью иметь для быстрого движения "только самое нужное", без лишнего, очень полезного, но тяжелого балласта. Такие системы, в противоположность предыдущим, не включают понятия иерархии в организации, не заботятся о какой-либо формализации потока работ. Их задача - обеспечить совместную работу людей в организации, даже если они разделены территориально, и сохранить результаты этой работы. Обычно реализованы в концепции "порталов". Они предоставляют сервисы хранения и публикации документов в интранет, поиска информации, обсуждения, средства назначения встреч. Такие системы находят заказчиков среди быстро развивающихся коммерческих компаний, рабочих групп в крупных фирмах и государственных структурах.

- Системы, имеющие развитые дополнительные сервисы. Например, сервис управления связями с клиентами (CRM - Customer Relation Management), управления проектами, биллинга, электронной почты и пр.

2.2 Обзор основных систем документооборота, представленных в России

В современной цифровой бизнес-среде комплексная программа управления документами с последовательными, безопасными и простыми процессами является важнейшим компонентом общего успеха любого бизнеса. Программные решения для управления документами предлагают мультидисциплинарный подход к управлению документами. Многие компании часто упускают из виду или недооценивают управление документами по разным причинам. В то время как

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		17

системный, хорошо спланированный процесс управления документами может повысить эффективность организации и сэкономить время и ресурсы, плохое управление документами может привести к неспособности принимать решения, раздражению сотрудников и отсутствию доступа к важной информации в нужное время, а также потеря бизнеса.

Последствия плохого управления документами:

- Несмотря на доступность инструментов и устройств, слишком многие предприятия плохо оснащены, чтобы справляться с невозможностью получить доступ к документам, иметь дело с неуместными файлами и т. Д. На повседневной основе, что делает ключевые ситуации очень неприятными для всех участников.

- Даже в учреждениях малого бизнеса, где не была проведена надлежащая оцифровка, уходит слишком много времени, просто сортируя его по картотечным шкафам. Более того, чтобы хранить свои документы, компаниям необходимо выделить свои ценные офисные помещения. Компании часто вынуждены платить большие деньги за хранение документов.

- Файлы неуместны, закопаны и потеряны;

- Связь между коллегами потеряна;

- Уровень стресса сотрудников повышается, поскольку их способность находить информацию и делиться ею становится совершенно сложной задачей.

Управление документами рассматривается как функция любого бизнеса, в котором используется надлежащая система управления документами для выполнения задач управления данными прямо с момента сбора, организации, управления и маршрутизации документов на протяжении их жизненных циклов, как правило, в системе, которая занимается со сложными неструктурированными записями. Таким образом, предприятия могут легко выполнять задачи, сокращать расходы, снижать риски и открывать конкурентные возможности.

Подводя итог, можно сказать, что управление документами важно для информационных работников, поскольку знания и решения, хранящиеся в документах, используются для достижения их целей. В эту мобильную, распределенную и ориентированную на клиентов эпоху критически важно хорошо

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		18

сотрудничать с коллегами, клиентами, партнерами и проверенными экспертами. Зарекомендовавшие себя программные решения для управления документами ориентированы на управление от небольшой группы до большой неструктурированной группы документов, жизненно важных для бизнеса.

Рынок также экспериментирует с новыми подходами, такими как открытый исходный код и облачные сервисы.

Ключевые преимущества:

- Архивирование тысяч и миллионов документов;
- Автоматизация офисов;
- Распределенное управление документами;
- Оцифровка и управление жизненным циклом всех документов;
- Легкое обнаружение документов;
- Доступность документов 24/7;
- Легко перемещать;
- Единая точка хранения.

При оценке необходимо учитывать следующие функциональные возможности, а также архитектуру программного обеспечения и масштабируемость. Но перед этим необходимо точно оценить потребности организации.

Основные критерии оценки программного обеспечения для управления документами:

- Отслеживание соответствия;
- Контрольный журнал;
- Архивирование документов;
- Хранение документов;
- Электронная подпись;
- Восстановление файлов;
- Преобразование типов файлов;
- Индексирование и классификация;
- Автономный доступ;

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		19

- Оптическое распознавание символов;
- Управление версиями;
- Инструменты для совместной работы;
- Безопасность;
- Поддержка нескольких языков.

2.2.1 OpenDocMan

OpenDocMan - это бесплатная веб-система управления документами с открытым исходным кодом (DMS). Это простое в использовании и настройке программное обеспечение позволяет пользователям добавлять в систему файлы любого типа. Программное обеспечение было выпущено под лицензией GPL. Он идеально подходит для частных лиц, малого бизнеса, некоммерческих организаций, учебных заведений, государственных учреждений и т. д.

Ключевая особенность:

- Возможность создания настраиваемых свойств документа;
- Мощный рабочий процесс управления документами;
- Автоматическая проверка документов;
- Автоматизированный процесс истечения срока действия файла;
- Возможность одобрить или отклонить новый документ;
- Параметры уведомления по электронной почте;
- Быстрый поиск по категории;
- Функция защищенного URL-адреса для скрытия параметров URL-адреса;
- Детализированный контроль доступа пользователей к каждому файлу;
- Контроль доступа для трех типов пользователей: пользователь, администратор и суперадминистратор;
- Поддерживает несколько языков, в том числе: китайский, английский, голландский, португальский, немецкий, хорватский, испанский и турецкий;
- Полное средство отчетности журнала аудита;
- Утверждение индивидуального рабочего процесса;

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		20

- Предварительный просмотр документа в браузере;
- Пакетная загрузка;
- Неограниченное количество пользователей;
- Пользователи могут получить поддержку от компании как по электронной почте, так и по телефону.

Управление документами:

- Добавить в систему любой тип файла;
- Не требуется FTP, можно загрузить прямо из используемого на рабочем месте браузера;
- Поля метаданных для каждого файла;
- Возможность назначить отдел / категорию каждому файлу;
- Функция извлечения для предотвращения перезаписи, правок;
- Лист регистраций изменений;
- Документы физически хранятся на сервере;
- Контроль срока действия файла;
- Создание настраиваемых свойств документа в соответствии с потребностями компании.

2.2.2 Document Pro

Document Pro - это веб-решение для управления / контроля документов с акцентом на требования, необходимые для достижения и поддержания сертификации FDA, QS / TS / и ISO, но с гибкостью и простотой использования для выполнения любых конкретных требований к контролю документов. Document Pro соответствует всем требованиям CFR, и позволяет отслеживать доступ сотрудников, редакции документов, запросы / утверждения на изменение и контролируемые документы в режиме реального времени и полностью онлайн.

Ключевая особенность:

- Интернет Доступ для клиентов и поставщиков;
- Целостность данных;

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		21

- Низкая совокупная стоимость владения;

- Гибкий дизайн хранилища документов;

В Document Pro заранее запрограммирована следующая базовая структура:

Первый уровень - Руководство по качеству;

Уровень два – Процессы;

Третий уровень - Рабочие инструкции;

Четвертый уровень - формы и контрольные списки;

Любое количество уровней, специфичных для организации.

Полнотекстовый поиск - Document Pro позволяет выполнять полнотекстовый поиск документов, чтобы точно определить данные, которые нужно увидеть.

Контроль версий.

Индивидуальный процесс утверждения - можно определять процесс утверждения документов, добавляя или меняя утверждающих.

Автоматическая маршрутизация утверждения - документы автоматически направляются для утверждения назначенным лицам, и для всех документов могут быть установлены разные классы маршрутизации.

Отслеживание утверждения - вы можете увидеть, где запрошенный документ находится в процессе утверждения.

Уведомление - лицо, назначенное утверждающим для документа, получает уведомление о том, что документ находится в очереди и требует его внимания.

Полные предыдущие версии документа хранятся в Document Pro с указанием даты редакции и доступны для просмотра в любое время. Система по умолчанию использует самую последнюю версию документа, чтобы избежать человеческой ошибки.

Безопасность соответствует паролю CFR Part 11 и повышенным требованиям безопасности. Отвечает требованиям CFR Part 11 к электронной подписи. Отвечает всем требованиям безопасности QS / TS для управления документами. Доступ пользователей может быть установлен для групп, отдельных лиц или документов, чтобы обеспечить точный доступ к вашим данным, который вам необходим.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		22

2.2.3 Другие системы

Указанные ниже системы в детальном описании повторялись бы с уже описанными, однако это не означает, что они уступают им.

Система Company Media разработана российской компанией "Интертраст" на основе Lotus Notes. Содержит широкий набор сервисов, поддерживающих делопроизводство, коллективное создание документов, контроль исполнения, управление договорами, управление проектами, управление персоналом, учет материальных ценностей и др. Сильной стороной является эффективная поддержка территориально распределенных структур управления за счет специальных методов, гарантирующих доставку заданий независимо от качества линий передач.

Продукт Staffware разработан компанией Staffware, распространяется в России фирмой "Весть-Метатехнология". Это серверная технология для управления потоками работ. Типичными пользователями Staffware могут стать телекоммуникационные компании, крупные и средние банки, гостиницы, другие организации.

"Эффект-Офис" - продукт петербургской компании "Гарант Интернешнл". При невысокой цене он достаточно функционален - содержит электронный архив, средства описания структуры организации, ограничения доступа по ролевому принципу и маршрутизации документов. Основная функция - электронный архив со средствами поиска информации. Кроме того, он включает в себя средства автоматизации делопроизводства, базирующиеся на технологиях маршрутизации документов контроля исполнения. В продукте реализована собственная электронная почта с поддержкой POP3/SMTP и UUCP. Отличительной чертой являются низкие требования к ресурсам оборудования и ориентация на небольшие организации (до 100-150 сотрудников).

Общая логическая схема системы электронного документооборота представлена на рисунке 2.1.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		23

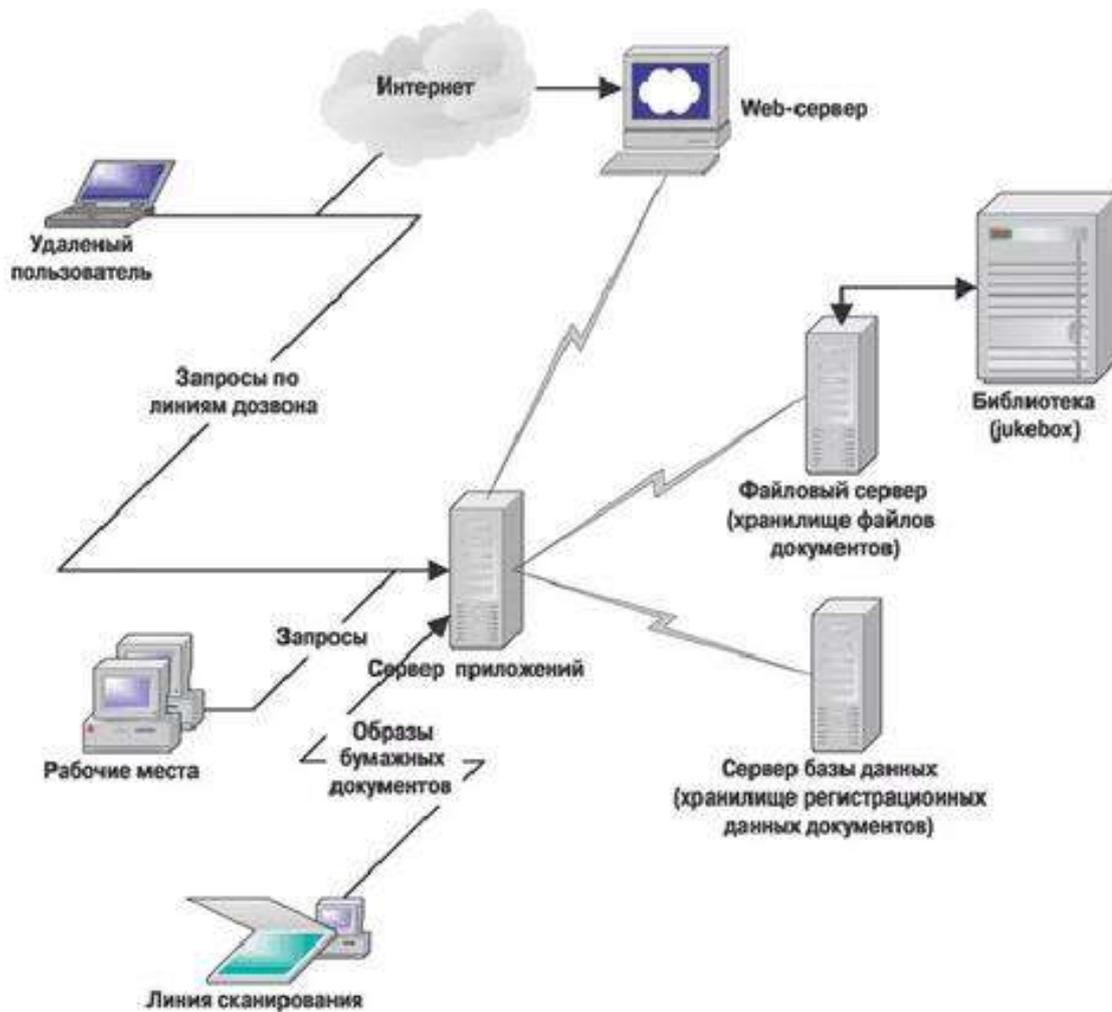


Рисунок 2.1 – Общая логическая схема системы электронного документооборота

Общая схема движения документа представлена на рисунке 2.2.

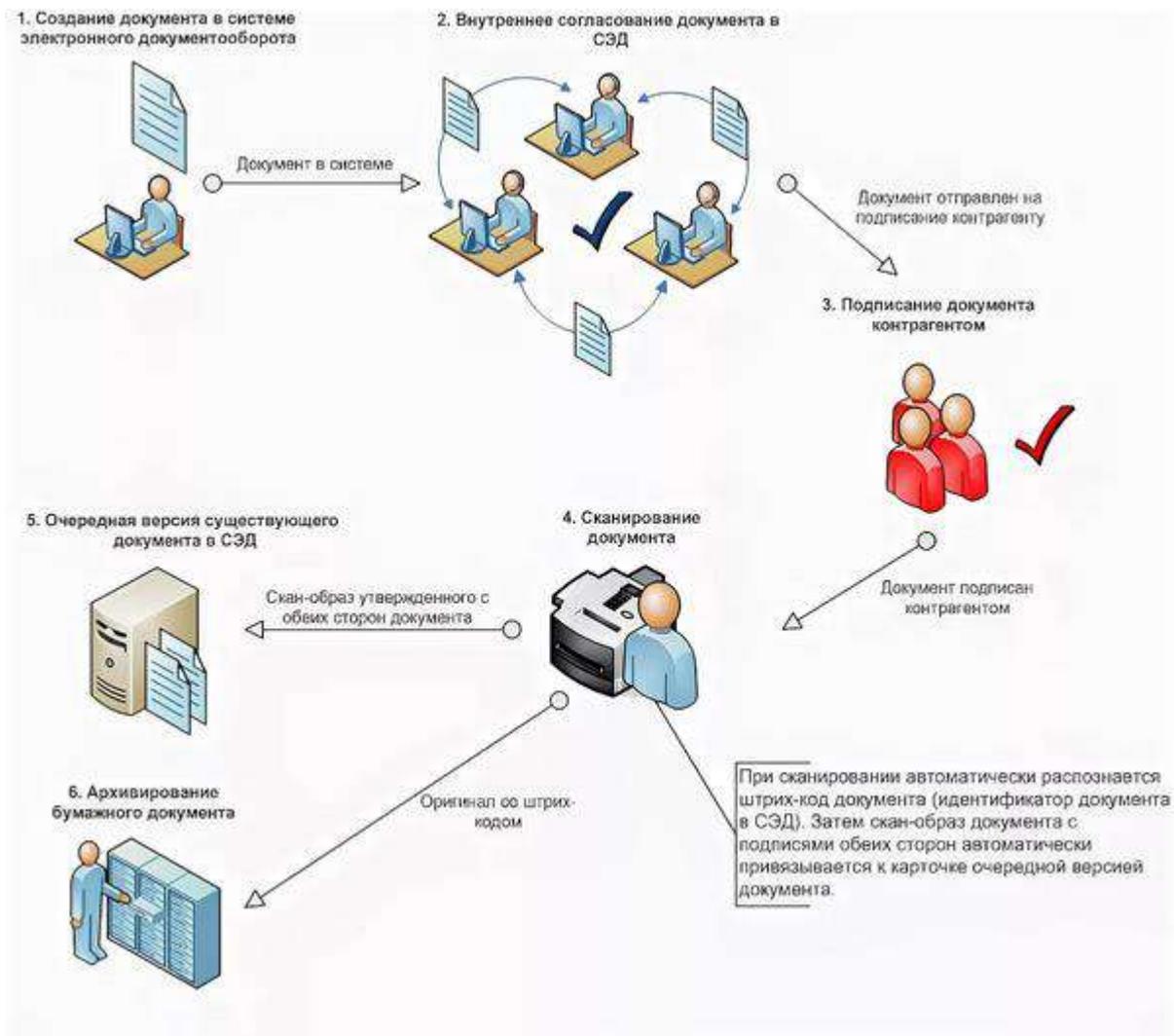


Рисунок 2.2 – Общая схема движения документа в системе электронного документооборота

2.3 Проблемы внедрения системы документооборота

Существуют общие практически для любых организаций проблемы, которые приходится решать при реализации, внедрении и сопровождении систем электронного документооборота.

- Консерватизм персонала, низкая образованность, нежелание обучаться и переобучаться. Боязнь прозрачности собственной деятельности для руководства, которая возникает после внедрения системы электронного документооборота.

- Фактор директора "советского типа" - нежелание непосредственно работать с компьютером, просматривать и редактировать документы.

- Постоянные структурные изменения в организации, слабая формализация бизнес-процессов.

- Необходимость обеспечения юридической силы документов.

- Необходимость взаимодействовать с внешним "бумажным" миром, в особенности если это касается параллельных структур в ассоциированных организациях или ведомствах, с которыми идет постоянная работа.

Кроме этих, существуют специфические для определенных типов организаций проблемы.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		26

3 ОБОСНОВАНИЕ НЕОБХОДИМОСТИ СОЗДАНИЯ НОВОГО ПО ДЛ ПРЕДПРИЯТИЯ И ПОСТАНОВКА ЗАДАЧИ

3.1 Термины, определения согласно стандарту, устанавливаемому организацию и порядок работы с претензиями потребителей

Претензия – устное или письменное требование, предъявляемое потребителем продавцу.

Потребитель – гражданин, имеющий намерение заказать или приобрести либо заказывающий, приобретающий или использующий товары исключительно для личных, семейных, домашних и иных нужд, не связанных с осуществлением предпринимательской деятельности.

Несоответствие – невыполнение требований.

Недостаток товара - несоответствие товара одному из условий:

- условиям договора;
- целям, для которых товар такого рода обычно используется;
- целям, о которых продавец был поставлен в известность потребителем при заключении договора;
- образцу и (или) описанию при продаже товара по образцу и (или) по описанию.

Существенный недостаток товара - неустранимый недостаток или недостаток, который не может быть устранен без несоразмерных расходов или затрат времени, или выявляется неоднократно, или проявляется вновь после его устранения.

3.2 Общие данные

Претензии потребителей могут содержать следующие виды требований:

- Перевод денег - перевод денежных средств с одного заказа на другой.
- Возврат переплаты – возврат излишне оплаченных денежных средств.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

- Замена товара - замена товара ненадлежащего качества на товар этой же марки (модели/артикула) или товар другой марки (модели/артикула) с перерасчетом покупной цены.

- Устранение недостатка товара - устранение недостатка товара путем замены отдельных элементов товара, доработки отдельных элементов товара, регулировки (настройки) и т. д.,

- Отказ от договора - отказ от исполнения договора в связи невозможностью исполнения договора продавцом, расторжение договора в связи с недостатком переданного товара, аннулирование заказа.

- Отказ от элементов товара – возврат денежных средств за неустановленные элементы товара при условии возврата элементов от потребителя.

- Неустойка - выплата неустойки в связи с нарушением сроков, предусмотренных договором, законом.

- Компенсация морального вреда – выплата денежных средств в связи с нравственными и/или физическими страданиями причинёнными ненадлежащим качеством товара и/или нарушением прав потребителя.

- Возмещение убытков – выплата денежных средств в целях возмещения расходов, понесенных покупателем в связи с ненадлежащим качеством товара и/или нарушением прав потребителя.

Претензии могут содержать несколько видов требований.

Претензии потребителей (в устной и письменной форме) могут поступать в салоны и (или) в головной офис компании. Все претензии потребителей, изложенные в письменной форме, подлежат регистрации. По претензиям потребителей, изложенным в устной форме, руководители салона (сети салонов) обязаны организовать встречу с потребителем (проведя при необходимости осмотр товара) с оформлением письменной претензии. В случае поступления устной претензии на предприятие любому работнику необходимо переадресовать потребителя на «горячую» линию (8-800.....).

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

Недостаточно только зарегистрировать претензию. Для обработки претензии необходимо выполнить несколько действий. Иногда эти действия могут выполняться разными сотрудниками.

Сотрудник, ответственный за работу с претензиями в салоне (сети салонов) обязан ежедневно:

- контролировать состояние претензий потребителей, размещенных в журнале «Претензии».

Для этого необходимо:

1. Выполнить сортировку претензий по типу и статусу;
2. Открыть список «Претензии покупателя», имеющих статус «Оформление»;

3. При выявлении в сформированном в соответствии с п.1-2 списке претензий, выделенных «красным цветом», сотрудник, ответственный за работу с претензиями в салоне (сети салонов), обязан в течение одних суток устранить замечания, зафиксированные на вкладке «Ошибки»;

- контролировать готовность ответов на претензии потребителей, размещенных в журнале «Претензии».

Для этого необходимо:

1. выполнить сортировку претензий по типу и статусу;
2. открыть список «Претензия покупателя», имеющих статус «Реализация решения»;

3. при наличии положительных решений по требованиям об устранении недостатков и наличии в текстовом поле указания на оформление рекламации, оформить на основании данного ЭД «Претензия» рекламацию;

4. При наличии на вкладке «Ответ» вложенных документов открыть скан-копию письма, распоряжения или текстовый документ соглашения и распечатать его (Соглашение в двух экземплярах).

5. При наличии скан-копии письма:

- а. в течение двух дней с момента перевода статуса претензии на «Реализация решения» ознакомить Потребителя с письмом под роспись;

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

- б. в течение двух дней с момента перевода статуса претензии на «Реализация решения» создать скан-копию ответа с подписью покупателя об ознакомлении;
- в. течение суток с момента ознакомления Покупателя с письмом вложить скан-копию с отметкой об ознакомлении на вкладку «Ответ» в ЭД «Претензия»;
- г. перевести (завершить) статус «Реализация решения».

Сотрудник, ответственный за работу с возвратами элементов из сети салонов, обязан ежедневно контролировать получение возвращаемых элементов.

Для этого необходимо:

- а. В программе КК открыть журнал распоряжений;
- б. Выполнить сортировку распоряжений по статусу;
- в. Открыть распоряжения на статусе «Подтверждение возврата»;
- г. При поступлении на участок возврата элементов, возвращаемых по распоряжению, проверить соответствие возвращаемого элемента элементу, выбранному в разделе «номенклатура» в распоряжении;
- д. При соответствии возвращаемого элемента номенклатуре - Открыть ЭД «Распоряжение» по возвращаемому элементу и перевести ЭД «Распоряжение» на статус «Направление на выплату»;
- е. Сохранить ЭД «Распоряжение».

Рассмотрение претензий отделом качества

1. Все претензии потребителей, находящиеся в журнале претензий на статусе «Регистрация», подлежат рассмотрению отделом контроля качества в течение двух суток.

2. При рассмотрении претензий сотрудник отдела качества должен выполнить следующие действия:

а. Проверить во вкладке «Приложения» наличие сопроводительной документации согласно списка, указанного в п.2 раздела «Оформление претензии в программе КК» настоящего стандарта;

б. При отсутствии необходимых документов указать об этом во вкладке «Ошибки» и «откатить» претензию на статус «Оформление». Строка претензии в журнале претензий при этом «заливается» красным цветом;

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		30

в. Определить список подразделений по заявленным требованиям;

г. Сформировать необходимое количество ЭД «Требование» с указанием в каждом ответственного подразделения, выбранного типа требования (в соответствии с ФЗ «О защите прав потребителей») и информацией к рассмотрению. При этом в качестве ответственного указать руководителя структурного подразделения;

д. В каждом ЭД «Требование» перевести статус на «Принятие решения», после чего ЭД «Требование» сохранить и закрыть;

е. Обновить претензию для формирования списка сформированных требований;

ж. Перевести статус ЭД «Претензия» на статус «Принятие решения», после чего сохранить и закрыть.

В случаях выявления при регистрации претензий нарушений, связанных с неправильным или неполным выполнением требований п. 5.2. настоящего стандарта (список претензий, находящихся на статусе «Оформление» и «залитых» красным цветом) сервисной службой предприятия еженедельно оформляется докладная записка руководителю предприятия для принятия мер дисциплинарного воздействия к руководителям сети салонов.

3.3 Сроки работы с претензиями

Отдел продаж (сеть салонов):

- сбор данных (проведение осмотра) и оформление претензии – 2 суток;
- оформление рекламации – 1 сутки;
- предоставление дополнительной информации и/или недостающей документации - 1 сутки;
- ознакомление с ответом, выплата денег по распоряжению – 2 суток.

Юридическая служба:

- рассмотрение данных и подготовка писем и/или распоряжений – 3 суток;
- архивирование, рассылка (работа с письмом, распоряжением) – 1 сутки.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		31

Отдел качества:

- формирование требований – 2 суток;
- рассмотрение сформированных требований – 3 суток.

Ответственные подразделения:

- рассмотрение сформированных требований – 5 суток.

Высшее руководство:

- рассмотрение и принятие решения – 1 сутки.

В результате анализа существующей системы работы с претензиями покупателей выявился ряд проблем:

- нет структурированного механизма ввода информации без ошибок;
- время ответа и качество проводимой работы с претензиями не адекватны требованиям потребителей;
- отсутствует системное решение вопросов накопления и повторного использования знаний по устранению причин;
- имеют место нескоординированные и неформализованные изменения, а также ответственные отделы действуют нескоординировано, что приводит к дублированию функций, а иногда и противоречивым действиям;
- нет доступной информации для менеджмента – для принятия обоснованного решения не хватает фактов, что существенно увеличивает сроки рассмотрения претензий.

3.4 Постановка задачи

Для достижения поставленной цели необходимо выполнить следующие задачи:

- выбор инструментальных средств создания программного обеспечения,
- разработка структуры базы данных,
- разработка структурной схемы и алгоритма программы,
- разработка интерфейса пользователя,
- создание программного кода.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		32

4 ТИПЫ И СРАВНЕНИЕ МОДЕЛЕЙ БАЗ ДАННЫХ

4.1 Иерархическая модель

Иерархическая модель базы данных, как следует из названия, представляет собой модель базы данных, в которой данные организованы в виде иерархического древовидного строения. Поскольку оно организовано на основе иерархии, каждая запись дерева данных должна иметь по крайней мере одного родителя, за исключением дочерних записей на последнем уровне, и каждый родитель должен иметь одну или несколько дочерних записей. Доступ к данным можно получить, следуя классифицированной структуре, всегда иницируемой корнем или первым родителем.

В этой модели данные хранятся в виде записей, которые представляют собой набор полей. Записи связаны с помощью ссылок, и тип записи указывает, какое поле содержится в записи. Каждое поле может содержать только одно значение. У него должен быть только один родитель для каждого дочернего узла, но у родительских узлов может быть более одного дочернего узла. Наличие нескольких родителей не допускается. Это основное различие между иерархической и сетевой моделью базы данных. Первый узел дерева называется корневым узлом. Когда необходимо получить данные, выполняется обход всего дерева, начиная с корневого узла. Эта модель представляет отношения «один ко многим».

4.2 Сетевая модель

Модель сетевой базы данных была продолжением иерархической модели базы данных и была разработана для решения некоторых проблем этой модели, в частности отсутствия гибкости. Вместо того, чтобы разрешать каждому ребенку иметь одного родителя, эта модель позволяет каждому ребенку иметь несколько родителей (она называет дочерних элементов членами, а родителей - владельцами).

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		33

Она обращается к необходимости моделировать более сложные отношения, такие как отношения «многие-ко-многим».

Конечно, у этой модели есть свои проблемы, иначе все бы ее использовали. Ее сложнее реализовать и поддерживать, и, хотя она более гибкая, чем иерархическая модель, она все еще имеет проблемы с гибкостью. Не все отношения могут быть удовлетворены путем назначения другого владельца, и программист все равно должен хорошо понимать структуру данных, чтобы сделать модель эффективной.

4.3 Реляционная модель

Модель реляционной базы данных управляется программным обеспечением, называемым системой управления реляционными базами данных, и к ней осуществляется доступ. Модель состоит из строк и столбцов, использующих SQL в качестве языка запросов для обслуживания и запросов к базам данных. Модель упорядочивает данные в таблицы по строкам и столбцам с отдельным ключом для поиска каждой строки. Строки определяются как кортежи или записи. Столбцы подразумеваются как атрибуты.

Обычно каждое отношение или таблица описывает один тип объекта, такой как продукт или покупатель. Если таблица описывает продукт, строка содержит название продукта, а столбец - поведение или описание продукта.

Для лучшего понимания модель реляционной базы данных используется как расширение базы данных, лежащей в основе реляционных значений. СУБД полностью основана на реляционной модели, широко применяемой во многих приложениях. Это распространенный выбор для хранения информации в базе данных, особенно при ведении финансовых записей, данных логистики, производственных продуктов, данных о персонале и т.д. Он заменил унаследованные иерархические базы данных и сетевые базы данных из-за своей простой реализации и функций администрирования. Первоначально СУБД сталкивалась с множеством сбоев, но после многих достижений, таких как

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		34

горизонтальное масштабирование кластеров и базы данных NoSQL, она стала применяться во многих надежных приложениях.

Отношение любого объекта определяется как массив определенных элементов, называемых кортежами, сгруппированными по схожим атрибутам. Кортеж обычно обозначает часть информации и описание объекта. Другими словами, кортеж описывается как уникальный, поэтому дублирование данных не обнаруживается. В основном объекты будут физическими элементами или отдельными усовершенствованиями концепции, которые используют такие функции, как select для поиска кортежей. Чтобы найти атрибуты, он выполняет команду проекта, а для объединения любых отношений использует команду соединения.

Отношения между базами данных можно изменить с помощью операторов вставки, обновления и удаления. Вновь созданные кортежи могут давать явные значения или извлекаться из запроса, в свою очередь, запросы могут также определять кортежи для удаления или обновления значений. Если какие-либо кортежи содержат первичный ключ, необязательно определять его как уникальный, поскольку атрибут кортежа уникален и состоит из суперключа. Внешний ключ используется для сопоставления основного столбца в таблице и в основном используется в отношениях перекрестных ссылок.

Домен объясняет набор возможных значений для данного атрибута и может быть ограничен значениями атрибута. Ограничение - это термин, который позволяет ограничить область действия атрибута. В основном он использует логические значения, чтобы указать, удовлетворяют ли данные ограничениям или нет. В основном это разворачивается в бизнес-политиках в базе данных на основе последующего использования данных на уровне приложений. Он также используется для остановки данных, которые уже хранятся в реляционных таблицах.

Индекс - это еще один метод обеспечения быстрого доступа к базе данных. Он создается на основе комбинации реляционных или атрибутов. При использовании индекса запросы, которые фильтруют атрибуты и идентифицируют

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		35

совпадающие corteжи, очень возможны без каких-либо индивидуальных проверок. Нормализация используется для предотвращения потери целостности данных и нарушений при манипулировании.

4.4 Сравнение трёх моделей

Иерархическая, позволяет быстро получать данные, но не отличается гибкостью. Иногда роль элемента (родителя или потомка) неясна и не подходит для иерархической модели.

Сетевая модель данных имеет более гибкую структуру, чем иерархическая, и поддерживает отношения «многие ко многим». Но быстро становится слишком сложной и неудобной для управления.

Реляционная модель более гибкая, чем иерархическая и проще для управления, чем сетевая, используется чаще всего.

Преимущества реляционной модели данных:

1. Простота использования;
2. Гибкость;
3. Независимость данных;
4. Безопасность;
5. Простота практического применения;
6. Слияние данных;
7. Целостность данных;

Недостатки:

1. Избыточность данных;
2. Низкая производительность.

4.5 Другие модели баз данных (ООСУБД)

Объектно-ориентированные базы данных - это разновидность системы управления базами данных. Объектно-ориентированные базы данных добавляют

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

функциональность базы данных к языкам объектного программирования, создавая более управляемые базы кода.

Объекты - это базовый строительный блок и экземпляр класса, тип которого является либо встроенным, либо определяемым пользователем.

Классы предоставляют схему или план для объектов, определяя поведение.

Методы определяют поведение класса.

Указатели помогают получить доступ к элементам базы данных объектов и установить отношения между объектами.

Основной характеристикой объектов в ООСУБД является возможность создания пользовательских типов. Объект, созданный в проекте или приложении, сохраняется в базе данных как есть. Объектно-ориентированные базы данных напрямую работают с данными как с законченными объектами. Вся информация поступает в один мгновенно доступный пакет объектов, а не в нескольких таблицах. Напротив, основные строительные блоки реляционных баз данных, таких как PostgreSQL или MySQL, представляют собой таблицы с действиями, основанными на логических связях между данными таблицы.

Объектно-ориентированные базы данных тесно связаны с концепциями объектно-ориентированного программирования. Четыре основных идеи объектно-ориентированного программирования:

- Полиморфизм;
- Наследование;
- Инкапсуляция;
- Абстракция.

Эти четыре атрибута описывают критические характеристики объектно-ориентированных систем управления.

Полиморфизм - это способность объекта принимать несколько форм. Эта возможность позволяет одному и тому же программному коду работать с разными типами данных.

Наследование создает иерархические отношения между связанными классами, делая части кода повторно используемыми. Определение новых типов

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		37

наследует все существующие поля и методы класса, а также расширяет их. Существующий класс является родительским, а дочерний класс расширяет родительский.

Инкапсуляция - это способность группировать данные и механизмы в один объект для обеспечения защиты доступа. В ходе этого процесса часть информации и детали того, как работает объект, скрываются, что обеспечивает безопасность данных и функций. Классы взаимодействуют друг с другом с помощью методов без необходимости знать, как работают определенные методы.

Абстракция - это процедура представления только основных характеристик данных для необходимой функциональности. Процесс выбирает важную информацию, в то время как ненужная информация остается скрытой. Абстракция помогает снизить сложность моделируемых данных и обеспечивает возможность повторного использования.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		38

5 СРАВНЕНИЕ И ВЫБОР СУБД

5.1 MySQL

Oracle MySQL Database Service - это полностью управляемая служба баз данных, которая позволяет разработчикам быстро разрабатывать и развертывать безопасные облачные приложения с использованием самой популярной в мире базы данных с открытым исходным кодом. Служба баз данных MySQL - единственная облачная служба MySQL со встроенным высокопроизводительным ускорителем запросов в памяти - HeatWave, который позволяет клиентам выполнять сложную аналитику непосредственно в отношении своих действующих баз данных MySQL, устраняя необходимость в сложных, трудоемких и трудоемких задачах. дорогостоящее перемещение данных и интеграция с отдельной аналитической базой данных. HeatWave на несколько порядков увеличивает производительность MySQL для аналитических и транзакционных запросов. Оптимизированная для Oracle Cloud Infrastructure (OCI) и доступная исключительно в ней, MySQL Database Service создается, управляется и поддерживается группами инженеров Oracle Cloud Infrastructure и MySQL.

5.2 PostgreSQL

PostgreSQL - это мощная объектно-реляционная база данных с открытым исходным кодом, которая использует и расширяет язык SQL в сочетании с множеством функций, обеспечивающих безопасное хранение и масштабирование самых сложных рабочих нагрузок с данными. Истоки PostgreSQL восходят к 1986 году в рамках проекта POSTGRES в Калифорнийском университете в Беркли и более 30 лет активно развиваются на базовой платформе. PostgreSQL заработал прочную репутацию благодаря своей проверенной архитектуре, надежности, целостности данных, надежному набору функций, расширяемости и приверженности сообщества разработчиков ПО с открытым исходным кодом к

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		39

последовательной разработке эффективных и инновационных решений. PostgreSQL работает во всех основных операционных системах, поддерживает ACID с 2001 года и имеет мощные надстройки, такие как популярный расширитель геопространственных баз данных PostGIS.

PostgreSQL пытается соответствовать стандарту SQL, где такое соответствие не противоречит традиционным функциям или может привести к неправильным архитектурным решениям. Поддерживаются многие функции, требуемые стандартом SQL, хотя иногда и с немного другим синтаксисом или функцией. Со временем можно ожидать дальнейшего продвижения к соответствию.

Различные функции PostgreSQL:

- Типы данных;
- Примитивы: целые, числовые, строковые, логические;
- Структурированный: дата / время, массив, диапазон, UUID;
- Документ: JSON / JSONB, XML, ключ-значение (Hstore);
- Геометрия: точка, линия, круг, многоугольник;
- Настройки: составные, пользовательские типы;
- Целостность данных УНИКАЛЬНЫЙ, НЕ ПУСТОЙ;
- Первичные ключи;
- Внешние ключи;
- Ограничения исключения;
- Индексирование: B-дерево, Многоколоночный, Выражения, Частичное;
- Расширенное индексирование: GiST, SP-Gist, KNN Gist, GIN, BRIN, индексы покрытия, фильтры Блума;
- Явные блокировки, консультативные блокировки;
- Параллелизм;
- Надежность, аварийное восстановление;
- Безопасность, надежная система контроля доступа;
- Поддержка международных наборов символов, например, через сопоставления ICU;

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		40

- Параметры сортировки без учета регистра и без учета диакритического знака;

- Полнотекстовый поиск.

Кроме того, PostgreSQL обладает высокой масштабируемостью как по огромному количеству данных, которыми он может управлять, так и по количеству одновременных пользователей, которые он может обслуживать. Существуют активные кластеры PostgreSQL в производственных средах, которые управляют многими терабайтами данных, и специализированные системы, управляющие петабайтами.

5.3 SQLite

Поскольку база данных SQLite представляет собой единый компактный файл в четко определенном кросс-платформенном формате, она часто используется в качестве контейнера для передачи содержимого из одной системы в другую. Отправитель собирает содержимое в файл базы данных SQLite, передает этот файл получателю, а затем получатель использует SQL для извлечения содержимого по мере необходимости.

База данных SQLite облегчает передачу данных между системами, даже если конечные точки имеют разные размеры слов и/или порядка байтов. Данные могут представлять собой сложное сочетание больших двоичных объектов, текста и небольших числовых или логических значений. Формат данных может быть легко расширен путем добавления новых таблиц и / или столбцов, не нарушая устаревшие приемники. Язык запросов SQL означает, что получатели не обязаны анализировать всю передачу сразу, а могут вместо этого запрашивать полученное содержимое по мере необходимости. Формат данных является "прозрачным" в том смысле, что он легко декодируется для просмотра человеком с использованием различных универсально доступных инструментов с открытым исходным кодом от нескольких поставщиков.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		41

Архив файлов, хранящихся в SQLite, только немного больше, а в некоторых случаях и меньше, чем эквивалентный ZIP-архив. И архив SQLite имеет инкрементное и атомарное обновление и возможность хранить гораздо более богатые метаданные.

SQLite-это хорошее решение для любой ситуации, которая требует объединения разнообразного контента в самодостаточный пакет с само описанием для пересылки по сети. Контент кодируется в четко определенном, кросс-платформенном и стабильном формате файла. Кодирование является эффективным, и приемники могут извлекать небольшие подмножества содержимого без необходимости читать и анализировать весь файл.

Архивы SQL полезны в качестве формата распространения обновлений программного обеспечения или контента.

Преимущества:

- файловая: вся база данных хранится в одном файле, что облегчает перемещение;
- использует SQL;
- является простой для работы с одним файлом и связанной библиотекой СИ.

Недостатки:

- отсутствие пользовательского управления;
- невозможность дополнительной настройки: SQLite нельзя сделать более производительной.

5.4 Выбор конкретной СУБД

Рассмотрев несколько СУБД, существующих на рынке и проанализировав все выше сказанное, для постановки и решения задачи автоматизации процесса обработки информации по претензиям и рекламациям на предприятии можно остановить выбор на СУБД SQLite.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		42

6 РЕАЛИЗАЦИЯ МОДЕЛИ БД, ОПИСАНИЕ РАБОТЫ ПРОГРАММЫ, РАЗРАБОТКА ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

6.1 Проектирование структуры базы данных

При проектировании структуры базы данных необходимо произвести выбор объектов, задать присущие им свойства, выявить связи, определить типы связей и необходимость взаимодействия объектов.

Создаваемая база данных будет состоять из 3-х основных таблиц.

Таблица 1 – Поля основной таблицы «Сотрудники»

Основная таблица	workers
ФИО сотрудника	workfullname
Имя пользователя	login
Пароль	password

Таблица 2 – Поля таблицы «Претензии»

Основная таблица	claim
Номер записи	id
Номер претензии	claimnumber
Название изделия	docname
Дата поступления претензии	receivdate
Список задач, требуемых выполнения	tasks
Количество листов	pages
Сроки выполнения задач	tasksdate
ФИО потребителя	cusfullname
Дата закрытия претензии	closedate
Статус	status

Таблица 3 – Поля таблицы «История активности»

Основная таблица	journal
Номер записи	id
Операция	operation
Дата	indate
Сотрудник	Ссылка на таблицу "Сотрудники"

Формирование каталога претензий происходит на локальном диске с доступом через базу данных. Каталог претензий и рекламаций представляет собой структурированную информацию о данных покупателей; о документах, которые поступают, регистрируются, рассматриваются, сопровождаются внутренними документами (инструкции, служебные записки, поручения, акты осмотра и т.д.); о статусе документов; о списках задач, требуемых выполнения; о сроках выполнения задач.

Ведение «Истории активности» позволяет проводить контрольные мероприятия по проверке наличия изменения или удаления документов, выявлению нарушений.

6.2 Реализация модели БД

Приложение создается в среде разработки Borland C++ Builder. Автоматически устанавливается библиотека BDE (Borland Database Engine).

BDE используется при работе с БД и обеспечивает для созданных приложений:

- непосредственный доступ к локальным базам данных;
- доступ к SQL-серверам с помощью драйверов Borland SQL Links;
- доступ к любым источникам данных, имеющим драйвер ODBC (Open DataBase Connectivity), например, к файлам электронных таблиц (Excel, Lotus 1-2-3), серверам баз данных, не имеющим драйверов SQL Links;

- создание приложений клиент-сервер, использующих разнородные данные;
- использование SQL (Structured Query Language - язык запросов к серверным СУБД), в том числе при работе с локальными данными;
- изоляцию приложения от средств языковой поддержки;
- изоляцию приложения от конфигурации системы и сети.

С помощью системных настроек возможно задать параметры системы, а также сети (таблица 4).

Таблица 4 – Параметры системы и сети, используемые библиотекой

Параметр	Описание	Значение по умолчанию
VERSION	версия BDE	
LOCAL SHARE	Если этот параметр равен true, можно использовать одни и те же данные приложениями, использующими BDE, и приложениями, не использующими BDE	FALSE
MINBUFSIZE	Минимальный размер буфера оперативной памяти для кэширования данных из БД. Возможные значения - от 32 до 65535 Кб, но не более, чем объем доступной Windows оперативной памяти	
MAXBUFSIZE	Максимальный размер буфера оперативной памяти для кэширования данных из БД. Должен быть выше, чем MINBUFSIZE, но не более, чем объем доступной Windows оперативной памяти. Должен быть кратен 128.	2048
LANGDRIVER	Языковой драйвер, соответствующий национальной версии операционной системы	
MAXFILEHANDLES	Максимальное число файлов, открываемых BDE. Может принимать целые значения от 5 до 256	48
SYSFLAGS	Внутренний параметр BDE	
LOW MEMORY USAGE LIMIT	Максимально допустимый объем основной памяти, используемый BDE	32
AUTO ODBC	если этот параметр принимает значение true, используются все ODBC-источники из файла ODBC.INI	FALSE
DEFAULT DRIVER	драйвер, используемый первым, если тип БД - FILE, и имя таблицы не имеет расширения.	
SQLQRYMODE	Метод исполнения запросов к серверам. Может принимать значения NULL, SERVER, LOCAL	NULL

Продолжение таблицы 4

SHAREDMEMSIZE	Максимальный размер для разделяемой оперативной памяти	2048
SHAREDMEMLOCATION	Предпочтительный адрес для размещения разделяемой области памяти.	E000 (Windows 7) 7000(Windows NT)

На странице Date возможно задать или изменить параметры дат (таблица 5).

Таблица 5 – Параметры настройки дат

Параметр	Описание	Значение по умолчанию
SEPARATOR	Символ-разделитель числа, месяца и года	Значение, содержащееся в настройках панели управления Windows
MODE	Параметр, определяющий порядок следования числа, месяца и года. Может принимать значения: 0(MDY), 1(DMY), 2(YMD).	Значение, содержащееся в настройках панели управления Windows
FOURDIGITYEAR	Параметр, определяющий отображаемое число цифр года. Может принимать значения: TRUE (4 цифры), FALSE (2 цифры)	TRUE
YEARBIASED	Параметр, определяющий, прибавлять или нет 1900 к значению года, если предыдущий параметр равен FALSE	TRUE
LEADINGZEROM	Параметр, определяющий, указывать ли лидирующие нули перед значением месяца, если оно является однозначным числом.	FALSE
LEADINGZEROD	Параметр, определяющий, указывать ли лидирующие нули перед значением числа, если оно является однозначным числом.	FALSE

На странице Time возможно задать или изменить параметры времени (таблица 6).

Таблица 6 – Параметры настройки времени

Параметр	Описание	Значение по умолчанию
TWELVEHOUR	Параметр, определяющий интервал отображения часов: 0-12 (TRUE) или 0-24 (FALSE)	TRUE
AMSTRING	Символьная строка для указания первой половины дня, если TWELVEHOUR=TRUE	AM
PMSTRING	Символьная строка для указания второй половины дня, если TWELVEHOUR=TRUE	PM
SECONDS	Параметр, определяющий, указывать ли секунды в значении времени	TRUE
MILSECONDS	Параметр, определяющий, указывать ли миллисекунды в значении времени .	FALSE

На странице Number возможно задать параметры числовых данных (таблица 7).

Таблица 7 – Параметры настройки чисел

Параметр	Описание	Значение по умолчанию
DECIMALSEPARATOR	Символ, отделяющий дробную часть числа	Значение, содержащееся в настройках панели управления Windows
THOUSANDSEPARATOR	Символ, отделяющий друг от друга "тройки" разрядов в многозначных числах	Значение, содержащееся в настройках панели управления Windows
DECIMALDIGITS	Максимальное число десятичных разрядов числа, полученного при преобразовании символьной строки	2
LEADINGZERON	Параметр, определяющий, указывать ли нуль перед дробной частью числа, чье абсолютное значение меньше единицы	TRUE

Создание таблиц в базе данных реализуется посредством программного кода.

Листинг программы создания таблиц в БД

```
#include "dbwork.h"

void CDataBase::db_open(char *filename)
{
    char *err=0;
    sqlite3_open(filename, &dbhandle);
```

```

        //sqlite3_exec(dbhandle, "ALTER TABLE 'works' ADD COLUMN
'repairedate' TEXT;", 0, 0, &err);
        sqlite3_exec(dbhandle, "\
CREATE TABLE 'docs' (\
    'id' INTEGER PRIMARY KEY AUTOINCREMENT,\
    'docnumber'          TEXT,\
    'docname'            TEXT,\
    'savedate'           TEXT,\
    'carrier'            TEXT,\
    'pages'              TEXT,\
    'outdate'            TEXT,\
    'employee'           TEXT,\
    'returndate'        TEXT,\
    'status'             TEXT \
);"
                                , 0, 0, &err);

        sqlite3_exec(dbhandle, "\
CREATE TABLE 'journal' (\
    'id' INTEGER PRIMARY KEY AUTOINCREMENT,\
    'docid'              TEXT,\
    'operation'          TEXT,\
    'date'               TEXT,\
    'employee'           TEXT\
);"
                                , 0, 0, &err);

        sqlite3_exec(dbhandle, "\
CREATE TABLE 'employees' (\
    'id' INTEGER PRIMARY KEY AUTOINCREMENT,\
    'name'               TEXT\
);"
                                , 0, 0, &err);

        sqlite3_exec(dbhandle, "\
                                CREATE TABLE 'users' (\
    'id' INTEGER PRIMARY KEY AUTOINCREMENT,\
    'name'               TEXT,\
    'password'           TEXT\
);"
                                , 0, 0, &err);
    }

void CDataBase::db_close()
{
    sqlite3_close(dbhandle);
}

void CDataBase::db_addnew()
{
    char *err=0;
    sqlite3_exec(dbhandle,

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		48

```

                                "INSERT INTO works (name)
VALUES ('test');",0,0,&err);
    }

    char *CDataBase::db_query(const char *sql,int
(*callback)(void*,int,char**,char**),void *data)
    {
        char *err=0;
        sqlite3_exec(dbhandle,
                    sql,callback,data,&err);
        return(err);
    }

int CDataBase::db_count(char *table,char * field)
{
    sqlite3_stmt *stmt;
    char temp[256];
    int result;

    sprintf(temp, "SELECT count(%) FROM %s ;",field,table);
    sqlite3_prepare_v2(dbhandle,temp,-1,&stmt,0);
    sqlite3_step(stmt);
    result = sqlite3_column_int(stmt,0);
    sqlite3_finalize(stmt);
    return(result);
}

int CDataBase::db_ispresent(char *table,char * field, char
*value)
{
    sqlite3_stmt *stmt;
    char temp[256];
    int rc,result;

    sprintf(temp, "SELECT %s FROM %s WHERE %s = '%s'
;",field,table,field,value);
    sqlite3_prepare_v2(dbhandle,temp,-1,&stmt,0);
    rc = sqlite3_step(stmt);
    result = (rc == 100);
    sqlite3_finalize(stmt);
    return(result);
}

int CDataBase::db_delete(char *table, int row)
{
    char *err=0;

    char temp[256];
    sprintf(temp, "DELETE FROM %s WHERE rowid = %d
;",table,row);
    sqlite3_exec(dbhandle,temp,0,0,&err);

    return(0);
}

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		49

```

int CDataBase::db_latest_value(char *table, char *field)
{
    sqlite3_stmt *stmt;
    char temp[256];
    int result;

    sprintf(temp, "SELECT %s FROM %s LIMIT 1 OFFSET %d
;",field,table,db_count(table,field)-1);
    sqlite3_prepare_v2(dbhandle,temp,-1,&stmt,0);
    sqlite3_step(stmt);
    result = sqlite3_column_int(stmt,0);
    sqlite3_finalize(stmt);
    return(result);
}

void CDataBase::db_stringfield(char *buf, char *table, char
*field, int rowid)
{
    sqlite3_stmt *stmt;
    char temp[256];
    char *text;

    sprintf(temp, "SELECT %s FROM %s WHERE rowid = %d
;",field,table,rowid);
    sqlite3_prepare_v2(dbhandle,temp,-1,&stmt,0);
    sqlite3_step(stmt);
    text = (char*) sqlite3_column_text(stmt,0);
    buf[0]=0;
    if(text && strcmp(text,"(null)"))
        strcpy(buf,text);
    sqlite3_finalize(stmt);
}

int CDataBase::db_field_by_field(char *table, char
*s_field,char *s_field_value, char *r_field,char *out_val)
{
    sqlite3_stmt *stmt;
    char temp[256];
    char *text;

    sprintf(temp, "SELECT %s FROM %s WHERE %s = '%s'
;",r_field,table,s_field,s_field_value);
    sqlite3_prepare_v2(dbhandle,temp,-1,&stmt,0);
    sqlite3_step(stmt);
    text = (char*) sqlite3_column_text(stmt,0);
    if(text)strcpy(out_val,text);
    sqlite3_finalize(stmt);
    if(text)
        return(0); //ok
    else
        return(-1); //error
}

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		50

Переход из программы во внешнее приложение для создания и редактирования документов:

```
//Функция срабатывает при нажатии на кнопку «Создать документ»
void __fastcall TNewOrderForm::FilesListDbClick(TObject
*Sender)
{
    HMODULE hModule = GetModuleHandleW(NULL); //Идентификатор
программы в системе Windows
    WCHAR temp[MAX_PATH]; //Временная переменная для хранения
полного пути к программе
    String path; //Полный путь до документа

    if(!FilesList->SelCount) return; //Если не выбран ни один
документ из списка, выход
    GetModuleFileNameW(hModule, temp, MAX_PATH); //Получаем
полное имя программы
    //например C:\Program Files\Docs\Claim.exe

    path=ExtractFileDir(temp)+"\files\\"+FilesList->Items-
>Strings[FilesList->ItemIndex];
    //ExtractFileDir(temp) = path
    //FilesList->Items->Strings[FilesList->ItemIndex] - имя
выбранного в ListBox файла, например claim1.docx
    //тогда path="C:\Program Files\Docs" + "\files\" +
"claim1.docx"
    //="C:\Program Files\Docs\files\claim1.docx"
    path.WideChar(temp,MAX_PATH); //Чтобы не было проблем с
русскими символами, меняем кодировку
    //Открываем файл в Windows, в нашем случае "C:\Program
Files\Docs\files\claim1.docx", откроется Word
    ShellExecute(Handle
,"open",path.c_str(),NULL,NULL,SW_SHOWNORMAL);

}
```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		51

6.3 Разработка интерфейса пользователя

Создание интерфейса разрабатываемого приложения начинается с оформления окна формы, добавления в нее компонентов. Формы являются основой приложений. Файл формы имеет расширение DFM, хранит изображение формы, а также ее свойства.

Просмотр и редактирование свойств объектов осуществляется через окно инспектора объектов. При открытии формы открывается окно редактора кода (рисунок 6.1).

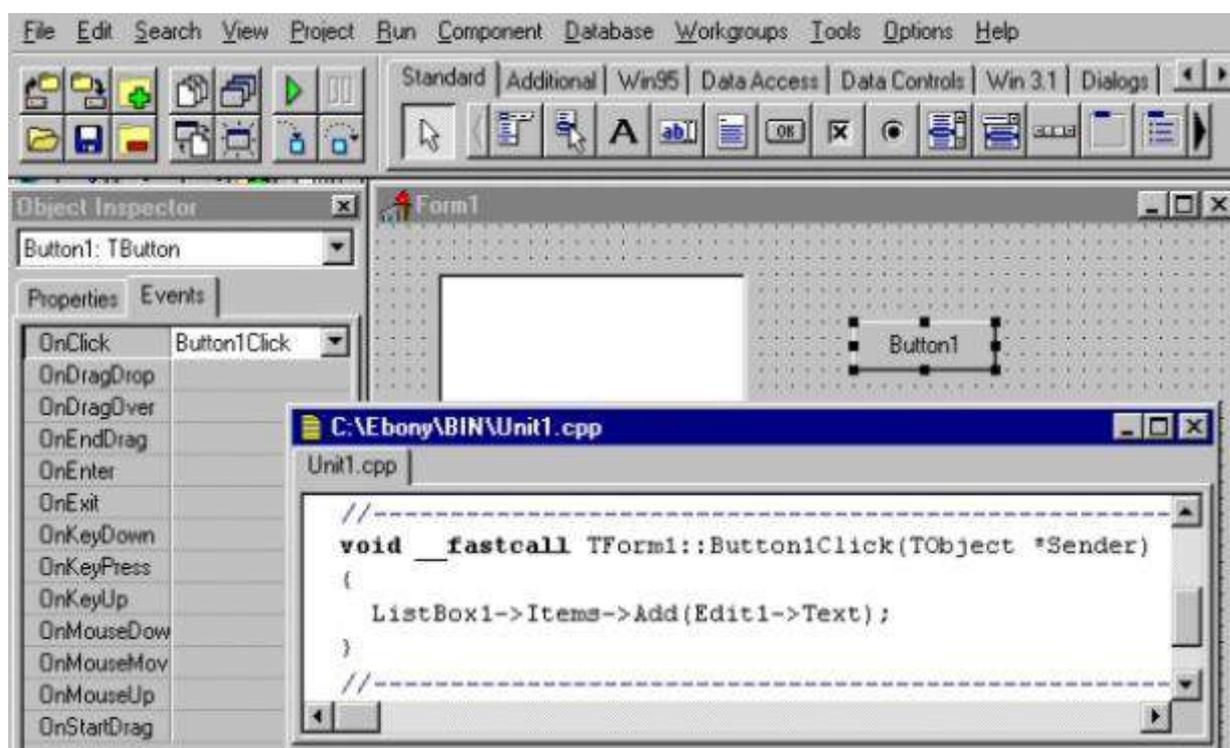


Рисунок 6.1 – Вид окна формы, окна инспектора объектов, панели компонентов и окна редактора кода

При создании формы, связанные с ней модуль и заголовочный файл с расширением *.h генерируются обязательно.

Файлы, образующие приложение - формы и модули - собраны в проект.

6.3.1 Создание окна входа в базу данных

Блок-схема алгоритма работы программы окна входа в программу показан на рисунке 6.2.

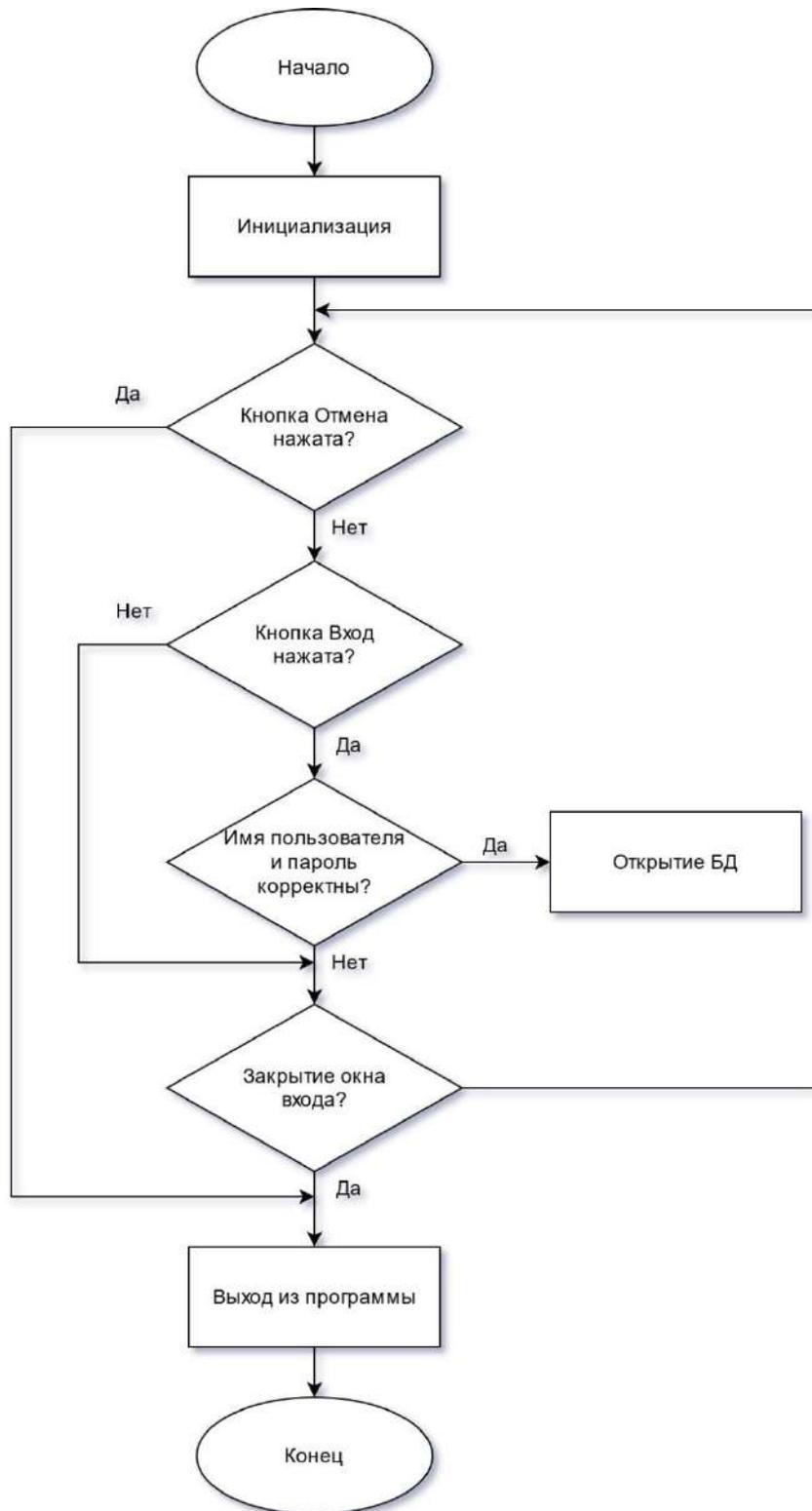


Рисунок 6.2 – Блок-схема алгоритма работы программы окна входа в программу

Листинг программы входа в БД

```
//-----  
-----  
#include <vcl.h>  
#pragma hdrstop  
  
#include "LoginUnit.h"  
#include "inifiles.hpp"  
#include "Main.h"  
#include "dbwork\dbwork.h"  
#include <memory>  
  
/* Класс для работы с базой данных */  
extern CDataBase DataBase;  
/* Объявление переменных */  
AnsiString CurrentUser;  
AnsiString path = ExtractFilePath(Application->ExeName);  
TIniFile *Ini = new TIniFile(path+"\\database.ini");  
  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TLoginForm *LoginForm;  
//-----  
-----  
__fastcall TLoginForm::TLoginForm(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
-----  
void __fastcall TLoginForm::CancelButtonClick(TObject *Sender)  
{  
    DataBase.db_close();  
    Application->Terminate();  
}
```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		54

```

//-----
-----
void __fastcall TLoginForm::EnterButtonClick(TObject *Sender)
{
    char temp[256];
    /* Проверка корректности имени пользователя и пароля */

    if(DataBase.db_count("users","name")==0)
    {
        //Новая база данных
        sprintf(temp, "INSERT INTO users (name,password) \
VALUES('%s','%s');",UserComboBox->Text.c_str(), PasswordEdit-
>Text.c_str());

        DataBase.db_query((char*)UTF8Encode(WideString(temp)).data(),0,0);
    }
    else
    {

        if (UserComboBox->Text == "admin" && PasswordEdit->Text ==
"00000")
        {
            status = authAdmin;
            ModalResult = mrOk;
        }
        else if (UserComboBox->Text == "Колесникова" && PasswordEdit-
>Text == "123123")
        {
            status = authUser;
            ModalResult = mrOk;
        }
        else if (UserComboBox->Text == "Уханова" && PasswordEdit-
>Text == "123456")
        {
            status = authUser;
            ModalResult = mrOk;
        }
    }
}

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		55

```

    }
    else if (UserComboBox->Text == "Лазарева" && PasswordEdit-
>Text == "123789")
    {
        status = authUser;
        ModalResult = mrOk;
    }
    else
        if(strcmp(temp,PasswordEdit->Text.c_str())) //Проверка
пароля
        {
            MessageBox(LoginForm->Handle,"Неверный
пароль!", "Database", 0);
            PasswordEdit->Clear();
            PasswordEdit->SetFocus();
            return;
        }
    else
        {

            ModalResult = mrCancel;
        }

}

Ini->WriteString("DATA","LastUserName",UserComboBox->Text);
CurrentUser=UserComboBox->Text;
MainForm->Show();
LoginForm->Hide();
}

// -----
-----

AuthStatus __fastcall TLoginForm::Execute()
{
    TLoginForm *frm = new TLoginForm(0);

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		56

```

    TLoginForm::status = authNone;
    frm->ShowModal();
    delete frm;
    return TLoginForm::status; // Возвращаем статус пользователя
}
//-----
int loginComboDBCallback(void *data, int argc, char **argv, char
**azColName)
{
    for(int i = 0; i<argc; i++)
    {
        LoginForm->UserComboBox->Items->Add(UTF8Decode(argv[i]));
    }
    return 0;
}

void __fastcall TLoginForm::FormCreate(TObject *Sender)
{
    /* Открываем базу данных */
    DataBase.db_open("database.sqlite");
}
//-----
void __fastcall TLoginForm::FormShow(TObject *Sender)
{
    LoginForm->Left=Screen->Width/2-LoginForm->Width/2;
    LoginForm->Top=Screen->Height/2-LoginForm->Height/2;
    PasswordEdit->Clear();
    PasswordEdit->SetFocus();
}
//-----

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		57

```

void __fastcall TLoginForm::PasswordEditKeyDown(TObject *Sender, WORD
&Key,
        TShiftState Shift)
{
    if(Key == 13)
        EnterButtonClick(Sender);
}
//-----
-----

void __fastcall TLoginForm::UserComboBoxSelect(TObject *Sender)
{
    PasswordEdit->Clear();
    PasswordEdit->SetFocus();
}
//-----
-----

```

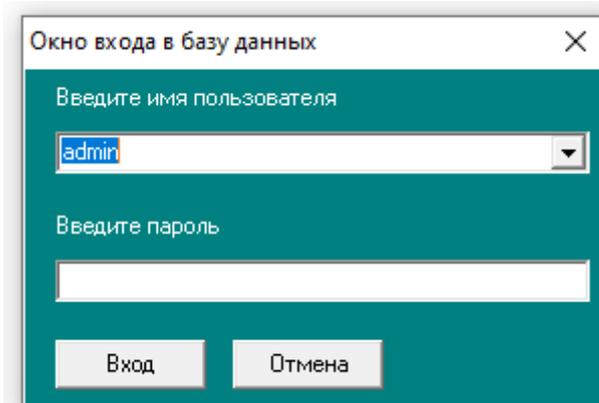


Рисунок 6.3 – Окно входа в базу данных

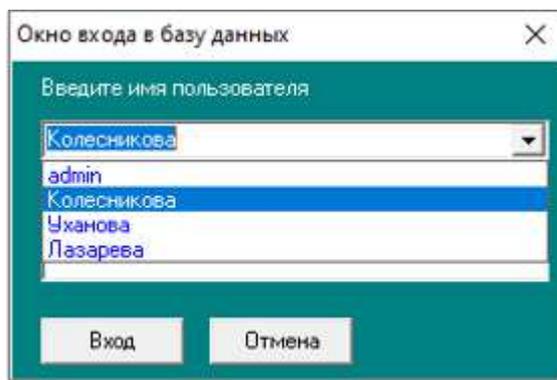


Рисунок 6.4 – Выбор пользователя в окне входа

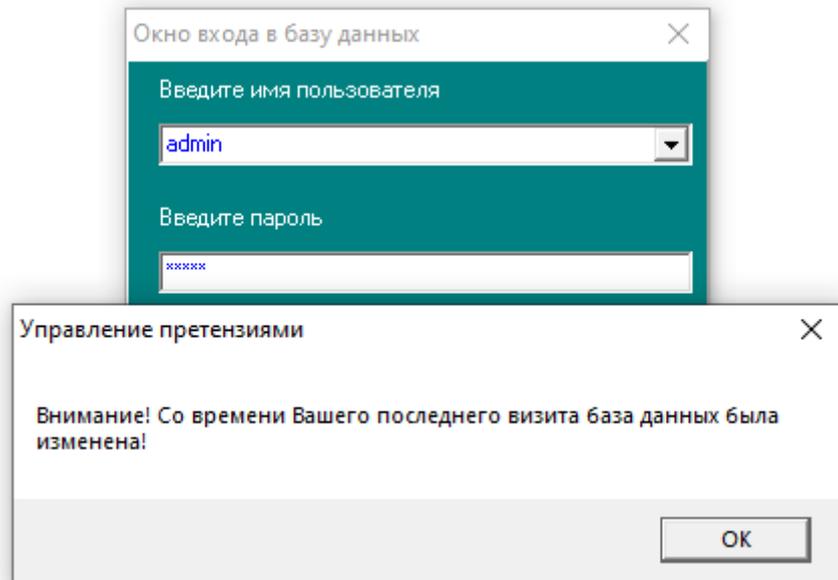


Рисунок 6.5 – Уведомление пользователя-администратора о внесении изменений в базу данных

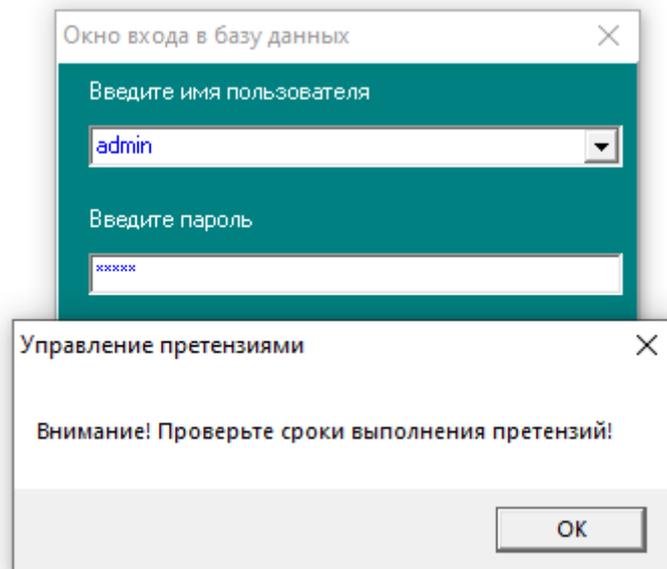


Рисунок 6.6 – Уведомление о проверке сроков

6.3.2 Создание основного окна базы данных

Блок-схема алгоритма основного цикла программы показана на рисунке 6.7.



Рисунок 6.7 – Блок-схема алгоритма основного цикла программы

Листинг основной программы

```
#include <vcl.h>
#pragma hdrstop

#include "inifiles.hpp"
#include "Main.h"
#include "Loginunit.h"
#include "NewDocUnit.h"
#include "OutUnit.h"
#include "DateUtils.hpp"
#include "retunit.h"
#include "journalUnit.h"
#include "dbwork\dbwork.h"
#include "changeUnit.h"
#include "ShellAPI.h"

//-----
-----

#pragma package(smart_init)
#pragma resource "*.dfm"

/* Объявление переменных */
TMainForm *MainForm;
CDataBase DataBase;
unsigned int lastRecordId;
extern TIniFile *Ini;
extern AnsiString CurrentUser;
bool outOfDate=false;
bool outOfSaveDate=false;

/* Функция поиска просроченных документов */
int CheckDateCallBack(void *data, int argc, char **argv, char
**azColName)
{
    int rowid=0;
    char temp[256];

    for(int i = 0; i<argc; i++)
    {
        if(!strcmp(azColName[i],"id"))
            rowid=StrToInt(UTF8Decode(argv[i]));
        else
        {
            if(!strcmp(azColName[i],"returndate"))
            {
                if(argv[i])
                    if(StrLen(argv[i]) &&
(CompareDate(Now(),StrToDate(UTF8Decode(argv[i]))) > 0))
                    {
                        sprintf(temp, "UPDATE docs SET status =
'Найдены просроченные документы' WHERE rowid = %d ;", rowid);

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		61

```

DataBase.db_query((char*)UTF8Encode(WideString(temp)).data(),0,0);
                                outOfDate=true;
                                }
                                }
                                }
                                }

return 0;
}

/* Обработка данных получаемых от СУБД при обновлении списка */
int UpdateListDBCallback(void *data, int argc, char **argv,
char **azColName)
{
    TListItem * Item;
    TStringList *list = new TStringList;
    String tmp,temp;
    TStringStream *richTextStream = new TStringStream("");

    lastRecordId=0;

    for(int i = 0; i<argc; i++)
    {
        if(!strcmp(azColName[i],"id"))
        { /* Создание столбцов для данных */
            Item = MainForm->MainListView->Items->Add();
            Item->Caption = UTF8Decode(argv[i]);
            if(lastRecordId<StrToInt(argv[i]))
lastRecordId=StrToInt(argv[i]);
                for(int k=0; k<MainForm->MainListView->Columns-
>Count; k++) Item->SubItems->Add("");

        }
        else
        {
            /* Распределение данных по столбцам */
            if(!strcmp(azColName[i],"status"))
                Item->SubItems->Strings[9] =
UTF8Decode(argv[i]);
            if(!strcmp(azColName[i],"id"))
                Item->SubItems->Strings[0] =
UTF8Decode(argv[i]);
            if(!strcmp(azColName[i],"tasks"))
                Item->SubItems->Strings[4] =
UTF8Decode(argv[i]);
            if(!strcmp(azColName[i],"claimnumber"))
                Item->SubItems->Strings[1] =
UTF8Decode(argv[i]);
            if(!strcmp(azColName[i],"pages"))
                Item->SubItems->Strings[5] =
UTF8Decode(argv[i]);
            if(!strcmp(azColName[i],"tasksdate"))

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		62

```

        Item->SubItems->Strings[6] =
UTF8Decode(argv[i]);
        if(!strcmp(azColName[i],"cusfullname"))
            Item->SubItems->Strings[7] =
UTF8Decode(argv[i]);
        if(!strcmp(azColName[i],"docname"))
            Item->SubItems->Strings[2] =
UTF8Decode(argv[i]) ;
        if(!strcmp(azColName[i],"receivdate"))
            Item->SubItems->Strings[3] =
UTF8Decode(argv[i]);
        if(!strcmp(azColName[i],"closedate"))
            Item->SubItems->Strings[8] =
UTF8Decode(argv[i]);
    }
}

delete(list);
delete richTextStream;
return 0;
}

//-----
-----
__fastcall TMainForm::TMainForm(TComponent* Owner)
: TForm(Owner)
{
}
//-----
-----

void __fastcall TMainForm::AddDocButtonClick(TObject *Sender)
{
    NewDocForm->ShowModal();
    UpdateListFromDB();
}
//-----
-----

void __fastcall TMainForm::FormClose(TObject *Sender,
TCloseAction &Action)
{
    /* Закрываем базу данных */
    DataBase.db_close();
    /* Сохранение форматирования окна программы */
    Ini->WriteInteger("LAYOUT","MainFormLeft",MainForm->Left);
    Ini->WriteInteger("LAYOUT","MainFormTop",MainForm->Top);
    Ini->WriteInteger("LAYOUT","MainFormWidth",MainForm-
>Width);
    Ini->WriteInteger("LAYOUT","MainFormHeight",MainForm-
>Height);

    Ini->WriteInteger("LAYOUT","JFormLeft",journalForm->Left);
    Ini->WriteInteger("LAYOUT","JFormTop",journalForm->Top);
    Ini->WriteInteger("LAYOUT","JFormWidth",journalForm-
>Width);
}

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		63

```

        Ini->WriteInteger("LAYOUT","JFormHeight",journalForm-
>Height);

        for(int i=0; i<MainListView->Columns->Count; i++)
            Ini-
>WriteInteger("LAYOUT","ColWidth"+IntToStr(i),MainListView-
>Column[i]->Width);
            for(int i=0; i<journalForm->journalList->Columns->Count;
i++)
                Ini-
>WriteInteger("LAYOUT","JColWidth"+IntToStr(i),journalForm-
>journalList->Column[i]->Width);
                Application->Terminate();
    }
    //-----
-----

void __fastcall TMainForm::FormShow(TObject *Sender)
{
    if(LoginForm->status == authAdmin)
        journalButton->Enabled = true;
    else
        journalButton->Enabled = false;

    /* Загрузка форматирования окна программы */
    MainForm->Left=Ini-
>ReadInteger("LAYOUT","MainFormLeft",100);
    MainForm->Top=Ini->ReadInteger("LAYOUT","MainFormTop",100);
    MainForm->Width=Ini-
>ReadInteger("LAYOUT","MainFormWidth",640);
    MainForm->Height=Ini-
>ReadInteger("LAYOUT","MainFormHeight",480);

    journalForm->Left=Ini-
>ReadInteger("LAYOUT","JFormLeft",100);
    journalForm->Top=Ini->ReadInteger("LAYOUT","JFormTop",100);
    journalForm->Width=Ini-
>ReadInteger("LAYOUT","JFormWidth",640);
    journalForm->Height=Ini-
>ReadInteger("LAYOUT","JFormHeight",480);

    for(int i=0; i<MainListView->Columns->Count; i++)
        MainListView->Column[i]->Width=Ini-
>ReadInteger("LAYOUT","ColWidth"+IntToStr(i),100);
    for(int i=0; i<journalForm->journalList->Columns->Count;
i++)
        journalForm->journalList->Column[i]->Width=Ini-
>ReadInteger("LAYOUT","JColWidth"+IntToStr(i),100);

    //if(outOfDate)

    /*Вывод уведомлений*/

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		64

```

        if(LoginForm->status == authAdmin) MessageBoxA(MainForm-
>Handle, "Внимание! Со времени Вашего последнего визита база данных
была изменена!",
        "Управление претензиями",0);
        //if(outOfSaveDate)
        if(LoginForm->status == authAdmin) MessageBoxA(MainForm-
>Handle, "Внимание! Проверьте сроки выполнения претензий!",
        "Управление претензиями",0);
    }
    //-----

```

```

void __fastcall TMainForm::N1Click(TObject *Sender)
{
    DataBase.db_close();
    Application->Terminate();
}
//-----

```

```

/* Функция обновления списка документов из БД */
void TMainForm::UpdateListFromDB()
{
    String filterstr[5];
    String query_string;
    int i=0;

    //DataBase.db_query((char*)UTF8Encode("SELECT * FROM docs
;").data(),CheckDateCallBack,0) ;

    if(filterComboBox->ItemIndex>0)
        query_string = "SELECT * FROM docs WHERE status = '"+
filterComboBox->Text +"'";
    else
        query_string = "SELECT * FROM docs ";

    if(SearchEdit->Text.Length())
        if(filterComboBox->ItemIndex>0)
            query_string=query_string+"OR employee LIKE '%"
+SearchEdit->Text + "%'";
        else
            query_string=query_string+"WHERE employee LIKE '%"
+SearchEdit->Text + "%'";
        else
            query_string=query_string+";";

    /* Очистка таблицы */
    MainForm->MainListView->Clear();
    /* Обновление данных из БД */
    MainForm->MainListView->Items->BeginUpdate();
}

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		65

```

DataBase.db_query((char*)UTF8Encode(query_string).data(),UpdateListDB
CallBack,0) ;
    MainForm->MainListView->Items->EndUpdate();

}

void __fastcall TMainForm::FormCreate(TObject *Sender)
{
    UpdateListFromDB();
}
//-----
-----

void __fastcall TMainForm::deleteDocButtonClick(TObject
*Sender)
{
    char temp[256];
    if(!MainListView->SelCount) return;
    sprintf(temp,"Действительно удалить
%s\r\n%s?",MainListView->Selected->Caption.c_str
        (),MainListView->Selected->SubItems-
>Strings[1].c_str());
    if(MessageBoxA(this->Handle,temp,"База данных",mbNo | mbNo)
!= 1) return;
    /* Удаление записи из БД */
    DataBase.db_delete("docs",MainListView->Selected-
>Caption.ToInt());
    UpdateListFromDB();
}
//-----
-----

void __fastcall TMainForm::MainListViewMouseDown(TObject
*Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    if(Button == mbRight)
        if(MainListView->SelCount == 1)
            /* ВЫЗОВ КОНТЕКСТНОГО МЕНЮ ПРИ НАЖАТИИ ПРАВОЙ КНОПКИ
МЫШИ */
                mainPopup->Popup(Mouse->CursorPos.x,Mouse-
>CursorPos.y);
}
//-----
-----

void __fastcall TMainForm::N3Click(TObject *Sender)
{
    deleteDocButton->Click();
}

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		66

```

//-----
-----

void __fastcall TMainForm::OutDocButtonClick(TObject *Sender)
{
    if(MainListView->SelCount)
    {
        OutDocForm->Execute(MainListView->Selected-
>Caption.ToInt());
        UpdateListFromDB();
    }
}
//-----
-----

```

```

void __fastcall TMainForm::filterComboBoxSelect(TObject
*Sender)
{
    UpdateListFromDB();
}
//-----
-----

```

```

void __fastcall TMainForm::SearchEditChange(TObject *Sender)
{
    UpdateListFromDB();
}
//-----
-----

```

```

void __fastcall TMainForm::ReturnDocButtonClick(TObject
*Sender)
{
    if(MainListView->SelCount)
    {
        retForm->Execute(MainListView->Selected-
>Caption.ToInt());
        UpdateListFromDB();
    }
}
//-----
-----

```

```

void __fastcall TMainForm::journalButtonClick(TObject *Sender)
{
    journalForm->ShowModal();
}
//-----
-----

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		67

```

void __fastcall TMainForm::N5Click(TObject *Sender)
{
    ReturnDocButtonClick(N5);
}
//-----

void __fastcall TMainForm::N4Click(TObject *Sender)
{
    OutDocButtonClick(N4);
}
//-----

void __fastcall TMainForm::EditDocButtonClick(TObject *Sender)
{
    if(MainListView->SelCount)
    {
        changeForm->Execute(MainListView->Selected-
>Caption.ToInt());
        UpdateListFromDB();
    }
}
//-----

void __fastcall TMainForm::N6Click(TObject *Sender)
{
    EditDocButtonClick(N6);
}
//-----

void __fastcall TMainForm::MainListViewDblClick(TObject
*Sender)
{
    OutDocButtonClick(N4);
}
//-----

void __fastcall TMainForm::N2Click(TObject *Sender)
{
    ShowMessage("База данных документации");
}
//-----

void __fastcall TMainForm::Button1Click(TObject *Sender)
{
    MainForm->Hide();
}

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		68

```

LoginForm->Show();
}
//-----
-----

void __fastcall TMainForm::MainListViewSelectItem(TObject
*Sender,
    TListItem *Item, bool Selected)
{
    retForm->Edit1->Text = "Нет";
    if(MainListView->ItemIndex == 5) retForm->Edit1->Text =
"Информационная справка по претензии";
    if(MainListView->ItemIndex == 7) retForm->Edit1->Text =
"Служебная записка";
    if(MainListView->ItemIndex == 6) retForm->Edit1->Text = "Акт
осмотра";
    if(MainListView->ItemIndex == 4) retForm->Edit1->Text =
"Уведомление о решении";
    if(MainListView->ItemIndex == 3) retForm->Edit1->Text =
"Служебная записка";

}
//-----
-----

void __fastcall TMainForm::Button2Click(TObject *Sender)
{
    HMODULE hModule = GetModuleHandleW(NULL);
    WCHAR temp[MAX_PATH];
    String path;

    GetModuleFileNameW(hModule, temp, MAX_PATH);

    path=ExtractFileDir(temp)+"\\Документ Microsoft Word.docx";
    path.WideChar(temp,MAX_PATH);

    ShellExecute(Handle
, "open", path.c_str(), NULL, NULL, SW_SHOWNORMAL);
}
//-----
-----

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		69

База данных

Файл О программе

Прием претензии | Рассмотрение / решение | Сопроводительные документы | Изменить | Удалить | История активности | Создать документ | Смена пользователя

Поиск по фамилии/дате: Фильтр по статусу:

Номер записи	Номер претензии	Название изделия	Дата поступления	Список задач	Листов	Сроки выполнения	ФИО потребителя	Дата закрытия	Статус
1	N°000337	К/л "Стафани"	02.11.2020		9		Молчанова О.И.	14.11.2020	Завершено
2	N°000785	К/л "Космо"	10.01.2021		8		Жазина Г.И.	03.02.2021	Завершено
3	N°000123	К/л "Канго"	28.02.2021		4		Кожинев П.А.	20.03.2021	Завершено
4	N°000674	К/л "Саванна"	15.03.2021	Передана в юридический отдел	4	15.05.2021	Перевалова С.В.		Перевод в рекламацию
5	N°000253	К/л "Прованс"	28.03.2021	Ожидание подтверждения выплаты клиенту	5	17.05.2021	Казакон Б.В.		Выполнение
6	N°000734	К/л "Эрика"	02.04.2021	Уведомить клиента о решении	6	15.05.2021	Ильиних В.Н.		Решение
7	N°000432	К/л "Ковадо"	10.05.2021	Требуется акт осмотра	3	16.05.2021	Пушкарева Н.Л.		Рассмотрение
8	N°000865	К/л "Элегант"	14.05.2021	Треб. исправления, дополнения	1	17.05.2021	Федосина К.С.		Регистрация

Рисунок 6.8 – Основное окно программы

Прием документа

Номер записи: Дата поступления:

Номер документа: ФИО потребителя:

Количество листов:

Название изделия:

Сохранить | Отмена

Рисунок 6.9 – Окно регистрации новой претензии

6.3.3 Создание окна «История активности пользователей» базы данных

Блок-схема алгоритма работы программы окна «История активности пользователей» показана на рисунке 6.10.



Рисунок 6.10 – Блок-схема алгоритма работы программы окна «История активности пользователей»

Листинг программы окна «История активности пользователей»

//-----

```
#include <vcl.h>
#pragma hdrstop
```

```
#include "journalUnit.h"
#include "dbwork\dbwork.h"
/* Класс для работы с базой данных */
extern CDataBase DataBase;
```

//-----

```
#pragma package(smart_init)
#pragma resource "*.dfm"
TjournalForm *journalForm;
```

```

//-----
-----
__fastcall TjournalForm::TjournalForm(TComponent* Owner)
    : TForm(Owner)
{
}

/* Обработка данных, полученных от СУБД */
int OpenJournalCallBack(void *data, int argc, char **argv, char
**azColName)
{
    TListItem * Item;
    TStringList *list = new TStringList;
    char temp[512];

    for(int i = 0; i<argc; i++)
    {
        if(!strcmp(azColName[i], "id"))
        { /* Создание столбцов для новой записи */
            Item = journalForm->journalList->Items->Add();
            Item->Caption = UTF8Decode(argv[i]);

            for(int k=0; k<journalForm->journalList->Columns-
>Count; k++) Item->SubItems->Add("");

        }
        else
        { /* Распределение данных по столбцам */
            if(!strcmp(azColName[i], "docid"))
            {
                Item->SubItems->Strings[0] =
UTF8Decode(argv[i]);

                DataBase.db_stringfield(temp, "docs", "docname", StrToInt(UTF8Decode(arg
v[i])));

                if(StrLen(temp))
                    Item->SubItems->Strings[1] = UTF8Decode(temp);
                else
                    Item->SubItems->Strings[1] = "id";
            }
            if(!strcmp(azColName[i], "operation"))
                Item->SubItems->Strings[2] =
UTF8Decode(argv[i]);
            if(!strcmp(azColName[i], "indate"))
                Item->SubItems->Strings[3] =
UTF8Decode(argv[i]);
            if(!strcmp(azColName[i], "employee"))
                Item->SubItems->Strings[4] =
UTF8Decode(argv[i]);
        }
    }

    delete(list);
    return 0;
}

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		72

```

}

//-----
void __fastcall TjournalForm::FormShow(TObject *Sender)
{
    char temp[256];
    /* Очистка окна журнала */
    journalList->Clear();
    /* Формирование запроса к базе данных */
    sprintf(temp, "SELECT * FROM journal ;");

    DataBase.db_query((char*)UTF8Encode(WideString(temp)).data(), OpenJournalCallBack, 0);
}
//-----

```

№ записи	Операция	Дата	Сотрудник
1	Вход	01.03.2021	Лазарева Е.И.
2	Вход	12.03.2021	Колесникова М.В.
3	Вход	13.03.2021	Лазарева Е.И.
4	Вход	20.03.2021	Лазарева Е.И.
5	Вход	01.04.2021	Колесникова М.В.
6	Вход	03.04.2021	Уханова О.В.
7	Вход	05.04.2021	Лазарева Е.И.
8	Вход	10.04.2021	Колесникова М.В.
9	Вход	15.04.2021	Лазарева Е.И.
10	Вход	21.04.2021	Уханова О.В.
11	Вход	11.05.2021	Лазарева Е.И.
12	Вход	15.05.2021	Колесникова М.В.
13	Вход	21.05.2021	Лазарева Е.И.
14	Вход	15.05.2021	Василенко А.А.

Рисунок 6.11 – Окно «История активности пользователей»

6.3.4 Создание окна «Сопроводительные документы» базы данных

Листинг программы создания окна «Сопроводительные документы»

```

/-----
-----

#include <vcl.h>
#pragma hdrstop

#include "RetUnit.h"
#include "dbwork\dbwork.h"
extern CDataBase DataBase;
//-----
-----

#pragma package(smart_init)
#pragma resource "*.dfm"

```

```

TretForm *retForm;

//-----
-----
__fastcall TretForm::TretForm(TComponent* Owner)
    : TForm(Owner)
{
}

int OpenWindowCallBack1(void *data, int argc, char **argv, char
**azColName)
{
    TListItem * Item;
    String tmp,temp;

    for(int i = 0; i<argc; i++)
    {
        if(!strcmp(azColName[i],"id"))
            retForm->IdEdit->Text = UTF8Decode(argv[i]);
        else
        {
            if(!strcmp(azColName[i],"docnumber"))
                retForm->docNumEdit->Text = UTF8Decode(argv[i]);
            if(!strcmp(azColName[i],"docname"))
                retForm->docNameEdit->Text = UTF8Decode(argv[i]);
            if(!strcmp(azColName[i],"outdate"))
                if(argv[i])
                    if(StrLen(argv[i]))
                        retForm->retDate->Date =
StrToDate(UTF8Decode(argv[i]));
            if(!strcmp(azColName[i],"employee"))
                retForm->employeeCombo->Text = UTF8Decode(argv[i]);
        }
    }

    return 0;
}

int retEmployeeCallback(void *data, int argc, char **argv, char
**azColName)
{
    for(int i = 0; i<argc; i++)
        retForm->employeeCombo->Items->Add(UTF8Decode(argv[i]));
    return 0;
}

void TretForm::Execute(unsigned int id)
{
    char temp[256];
    employeeCombo->Clear();
    DataBase.db_query((char*)UTF8Encode("SELECT name FROM employees
;").data(),retEmployeeCallback,0);
    employeeCombo->Text = "";
    sprintf(temp,"SELECT * FROM docs where id = '%d';",id);
}

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		74

```

DataBase.db_query((char*)UTF8Encode(WideString(temp)).data(),OpenWindowCallback1,0);
    if(employeeCombo->Text.Length()) this->ShowModal();
}
//-----
-----
void __fastcall TretForm::Button1Click(TObject *Sender)
{
    char temp[256];

    this->Close();
    return;

    if(employeeCombo->Text.Length()<2)
    {
        ShowMessage("Необходимо ввести имя сотрудника!");
        return;
    }

    if(!DataBase.db_ispresent("employees","name",(char*)UTF8Encode(employeeCombo->Text.c_str()).data()))
    {
        sprintf(temp,"Сотрудника %s нет в базе.\r\nДобавить %s в базу данных?",employeeCombo->Text.c_str(),employeeCombo->Text.c_str());
        if(MessageBoxA(this->Handle,temp,"База данных",mbNo | mbYes) != 1) return;
        sprintf(temp, "INSERT INTO employees (name) \
VALUES ('%s');",
                employeeCombo->Text.c_str());

        DataBase.db_query((char*)UTF8Encode(WideString(temp)).data(),0,0);
    }
    sprintf(temp, "UPDATE docs SET outdate = '', \
employee = '', returndate = '', status = '%s' WHERE rowid = %s ;"
            ,"B архиве",IdEdit->Text.c_str());

    DataBase.db_query((char*)UTF8Encode(WideString(temp)).data(),0,0);

    sprintf(temp, "INSERT INTO journal (docid, operation, date, employee) \
VALUES ('%s', '%s', '%s', '%s');",
            IdEdit->Text.c_str(), "Возврат",
            Date().CurrentDate().DateString().c_str(),employeeCombo->Text.c_str());

    DataBase.db_query((char*)UTF8Encode(WideString(temp)).data(),0,0);

    this->Close();
}
//-----
-----

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		75

```

void __fastcall TretForm::Button2Click(TObject *Sender)
{
    this->Close();
}
//-----

```

Рисунок 6.12 – Окно «Сопроводительные документы»

6.3.5 Создание окна изменения документа в базе данных

Листинг программы создания окна «Изменить»

```

//-----
-----

#include <vcl.h>
#pragma hdrstop
#include "dbwork\dbwork.h"
extern CDataBase DataBase;
#include "changeUnit.h"
//-----
-----

#pragma package(smart_init)
#pragma resource "*.dfm"
TchangeForm *changeForm;
//-----
-----

__fastcall TchangeForm::TchangeForm(TComponent* Owner)
: TForm(Owner)
{
}
//-----
-----

int ChangeEmployeeCallback(void *data, int argc, char **argv, char
**azColName)
{
    for(int i = 0; i<argc; i++)

```

```

        changeForm->employeeCombo->Items-
>Add(UTF8Decode(argv[i]));
    return 0;
}

int OpenChangeCallBack(void *data, int argc, char **argv, char
**azColName)
{
    TListItem * Item;
    String tmp,temp;

    for(int i = 0; i<argc; i++)
    {
        if(!strcmp(azColName[i],"id"))
            changeForm->IdEdit->Text = UTF8Decode(argv[i]);
        else
        {
            if(!strcmp(azColName[i],"docnumber"))
                changeForm->docNumEdit->Text = UTF8Decode(argv[i]);
            if(!strcmp(azColName[i],"pages"))
                changeForm->pagesEdit->Text = UTF8Decode(argv[i]);
            if(!strcmp(azColName[i],"docname"))
                changeForm->docNameEdit->Text = UTF8Decode(argv[i]);
            if(!strcmp(azColName[i],"employee"))
                changeForm->employeeCombo->Text =
UTF8Decode(argv[i]);
            if(!strcmp(azColName[i],"savedate"))
                changeForm->saveDate->Date =
StrToDate(UTF8Decode(argv[i])) ;
            if(!strcmp(azColName[i],"carrier"))
                changeForm->carrierCombo->Text =
UTF8Decode(argv[i]);
        }
    }

    return 0;
}

void __fastcall TchangeForm::Button1Click(TObject *Sender)
{
    char temp[256];

    if(docNameEdit->Text.Length()<2)
    {
        ShowMessage("Необходимо ввести название документа!");
        return;
    }

    sprintf(temp, "UPDATE docs SET docnumber='%s', docname='%s',
savedate='%s', carrier='%s', \
pages='%s' WHERE rowid = %s ;",
        docNumEdit->Text,docNameEdit->Text, saveDate-
>Date.DateString().c_str(),

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		77

```

        carrierCombo->Text,pagesEdit->Text.c_str(), IdEdit-
>Text.c_str());

DataBase.db_query((char*)UTF8Encode(WideString(temp)).data(),0,0);

    sprintf(temp, "INSERT INTO journal (docid, operation, date,
employee) \
        VALUES('%s','%s','%s','%s');",
        IdEdit->Text.c_str(), "ИЗМЕНЕН",
Date().CurrentDate().DateString().c_str(),employeeCombo-
>Text.c_str());

DataBase.db_query((char*)UTF8Encode(WideString(temp)).data(),0,0);

    this->Close();
}
//-----
-----

void TchangeForm::Execute(unsigned int id)
{
    char temp[256];

    sprintf(temp,"SELECT * FROM docs where id = '%d';",id);

    DataBase.db_query((char*)UTF8Encode(WideString(temp)).data(),OpenChan
geCallBack,0);

    //employeeCombo->Clear();
    // DataBase.db_query((char*)UTF8Encode("SELECT name FROM employees
;").data(),ChangeEmployeeCallback,0);
    //employeeCombo->Text = "";
    this->ShowModal();
}

void __fastcall TchangeForm::Button2Click(TObject *Sender)
{
    this->Close();
}
//-----
-----

```

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		78

6.4 Описание программы

Функциональная схема программы представлена на рисунке 6.13.

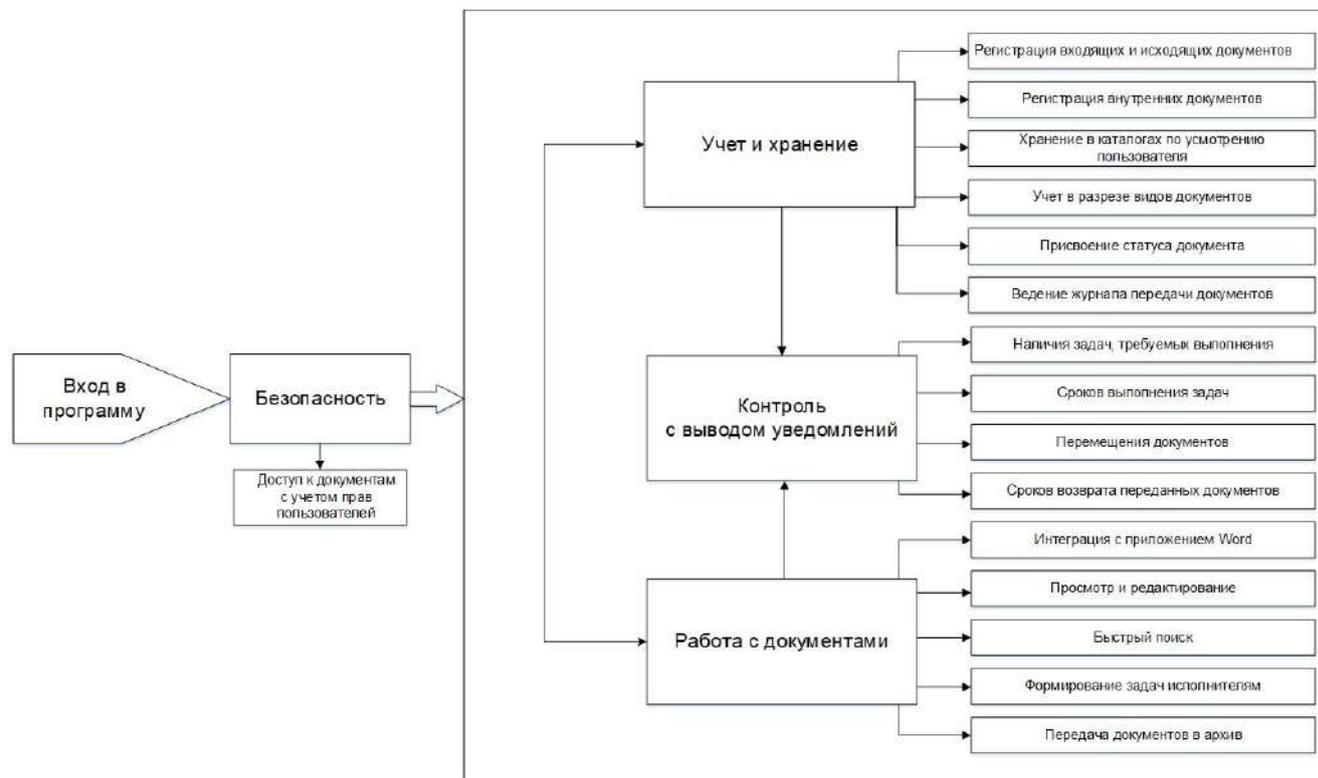


Рисунок 6.13 – Функциональная схема программы

6.5 Поставка созданного приложения

Одной из важнейших проблем при разработке заказных программных продуктов является окончательная сборка проекта и поставка приложений. При окончательной сборке исполняемого файла (то есть непосредственно перед созданием дистрибутива) следует убрать из исполняемого файла отладочную информацию. Для этого следует из менеджера проектов вызвать диалоговую панель Project Options и нажать кнопку Release.

Если приложение не использует базы данных, в большинстве случаев будет достаточно просто скопировать на компьютер пользователя исполняемый файл и другие входящие в комплект поставки файлы (например, файл справки, файл оглавления справки и др.).

Если приложение использует базы данных, следует, помимо приложения, установить на компьютер пользователя библиотеку Borland Database Engine. Установка этой библиотеки заключается в копировании файлов этой библиотеки на компьютер пользователя, внесении сведений о ней в реестр (раздел HKEY_LOCAL_MACHINE//SOFTWARE//Borland//Database Engine), установке пиктограммы для утилиты конфигурации BDE, а также настройке псевдонимов, необходимых для корректной работы данного приложения.

Следует также установить на компьютер пользователя необходимые драйверы SQL Links для доступа к серверным СУБД, если они используются, а также клиентское программное обеспечение для работы с соответствующей серверной СУБД (например, SQL*Net для Oracle). Кроме того, пользователь должен обладать необходимыми привилегиями для доступа к используемым вашим приложением объектам базы данных - таблицам, хранимым процедурам и др.

Создание дистрибутива с помощью InstallShield Express показано на рисунке 6.14.

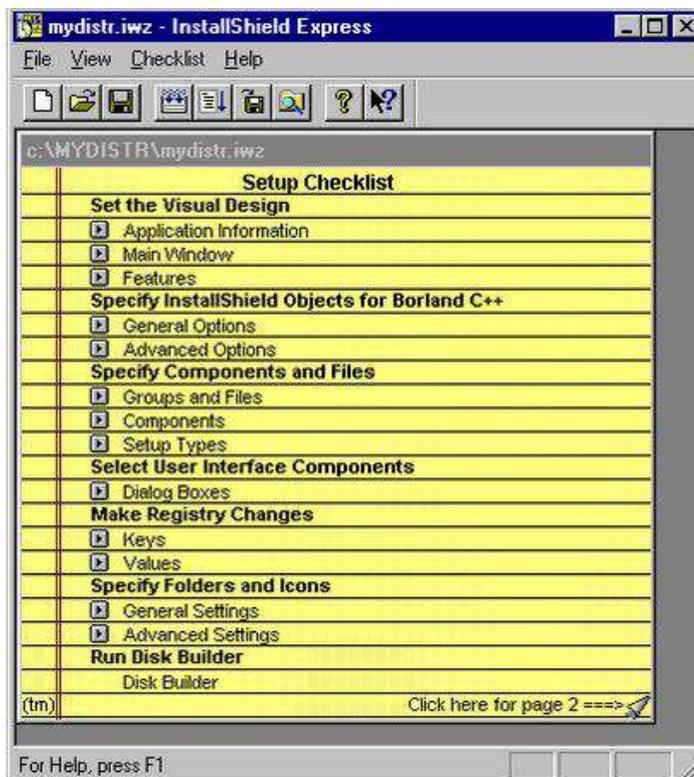


Рисунок 6.14 – Главный экран InstallShield Express

В разделе Set the Visual Design следует заполнить поля со сведениями о приложении.

Install Shield содержит несколько переменных, идентифицирующих диски и каталоги компьютера пользователя, что позволяет не знать реальную систему каталогов и дисков этого компьютера:

- <INSTALLDIR> - каталог для установки, указанный пользователем;
- <WINDIR> - каталог, в котором содержится Windows;
- <WINSYSDIR> - каталог Windows\System;
- <WINDISK> - диск, на котором содержится Windows;
- <WINSYSDISK> - диск, на котором содержится каталог Windows\System;
- <ProgramFilesDir> - каталог Program Files.

6.6 Перенос созданного приложения в архитектуру клиент-сервер

Способов переноса данных может быть несколько. Первый, самый примитивный, и, к сожалению, весьма распространенный, представляет собой перенос структур данных и самих данных "как есть" с одной платформы на другую с заменой типов данных настольной СУБД на эквивалентные им типы данных сервера. Этой цели служит утилита Data Migration Wizard, входящая в комплект поставки C++Builder.

Достоинством этого метода переноса данных является его простота. Но при его применении далеко не все преимущества архитектуры клиент/сервер оказываются использованными. Скорость выполнения запросов к данным, хранящимся теперь в серверной базе данных, действительно может возрасти - их будет выполнять сервер. Однако, перенося данные таким способом, на сервере создается только то, что было в исходной базе данных - таблицы и индексы. Новые

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		81

объекты серверной базы данных, реализующие бизнес-правила, такие, как триггеры и хранимые процедуры, при этом созданы не будут - их придется создавать вручную, занимаясь кодированием на процедурном расширении SQL, характерном для данного сервера, либо, как и в случае настольных СУБД, описывать их внутри клиентского приложения, рискуя нарушить ссылочную целостность данных.

Второй способ базируется на использовании CASE-средств (CASE расшифровывается как Computer-Aided System Engineering) для восстановления схемы базы данных по имеющимся таблицам (так называемого обратного проектирования), замены платформы и описания связей между таблицами с точки зрения реакции сервера на попытки того или иного изменения данных со стороны клиентского приложения.

Для переноса данных с одной платформы на другую обычно используется компонент TBatchMove. Этот компонент обеспечивает копирование данных из одной таблицы в другую. Основные свойства этого компонента следующие: Source – таблица (или запрос), откуда копируются данные, Destination – таблица, куда копируются данные, Mapping – определяет соответствие между колонками исходной и результирующей таблиц (для идентичных таблиц это свойство определять не обязательно), Mode – тип перемещения (batAppend – добавляет новые строки в результирующую таблицу, batUpdate – заменяет строки в результирующей таблице на соответствующие строки оригинала, batCopy – копирует строки в результирующую таблицу, переписывая ее, batDelete – удаляет записи в результирующей таблице, соответствующие записям оригинала), KeyViolTableName и ProblemTableName – имена дополнительных таблиц для помещения записей, чье копирование запрещено правилами ссылочной целостности или по каким-либо причинам невозможно (например, из-за несоответствия типов данных), ChangedTableName – имя таблицы для помещения измененных записей. Копирование данных происходит при выполнении метода Execute(). Этот метод может быть вызван непосредственно из среды разработки с помощью контекстного меню компонента TBatchMove.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		82

ЗАКЛЮЧЕНИЕ

В ходе написания работы было разработано программное обеспечение системы обработки претензий и рекламаций для производственного предприятия, разработаны структура базы данных и интерфейс доступа конечного пользователя. Программное обеспечение реализовано на языке C++ и представляет собой исполняемый файл для систем Microsoft Windows. Приложение автоматизирует процесс доступа и управления информацией в базе данных с заданной структурой, обеспечивает контроль наличия задач, требуемых выполнения, сроков выполнения задач с выводом уведомлений на экран.

Структура разработанной базы данных состоит из 3 таблиц: «Сотрудники», «Претензии» и «История активности».

Таблицы полностью включают в себя информацию о работе отдела качества, связь между таблицами реализована программно.

Таким образом, разработанное программное обеспечение в полной мере удовлетворяет требованиям предприятия к базе данных отдела качества, учитывает все движения и основные направления деятельности отдела, учитывает специфику его работы. При этом база данных крайне проста в применении, имеет удобный интерфейс работы и понятные дополнительные операции.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		83

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Голицына, О. Л. Базы данных / О.Л. Голицына, Н.В. Максимов, И.И. Попов. – М.: Форум, 2018. – 400 с.

2 Кириллов, В.В. Введение в реляционные базы данных (+ CD-ROM) / В.В. Кириллов. – М.: БХВ-Петербург, 2019. – 318 с.

3 Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли. – М.: Вильямс И.Д., 2017. – 1440 с.

4 Кузнецов, С. Д. Базы данных. Модели и языки / С.Д. Кузнецов. – М.: Бином-Пресс, 2016. – 720 с.

5 Макин, Дж.К. Проектирование серверной инфраструктуры баз данных Microsoft SQL Server 2005 / Дж.К. Макин. – М.: Русская редакция, 2016. – 560 с.

6 Мартишин, С. А. Базы данных. Практическое примечание СУБД SQL и NoSQL. Учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. – М.: Форум, Инфра-М, 2016. – 368 с.

7 Мартишин, С.А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: Методы и средства проектирования информационных систем и технолог / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. – М.: Форум, 2017. – 62 с.

8 Семакин, И.Г. Основы алгоритмизации и программирования: Учебник / И.Г. Семакин. – М.: Academia, 2017. – 384 с.

9 Стружкин, Н.П. Базы данных: проектирование. Учебник для академического бакалавриата / Н.П. Стружкин, В.В. Годин. – Люберцы: Юрайт, 2016. – 477 с.

10 Стружкин, Н.П. Базы данных: проектирование. Практикум: Учебное пособие для академического бакалавриата / Н.П. Стружкин, В.В. Годин. – Люберцы: Юрайт, 2016. – 291 с.

11 Хомоненко, А. Работа с базами данных в C++ BUILDER / А. Хомоненко. – М.: Книга по Требованию, 2017. – 488 с.

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		84

12 Черпаков, И.В. Основы программирования: Учебник и практикум для прикладного бакалавриата / И.В. Черпаков. – Люберцы: Юрайт, 2016. – 219 с.

13 Документация по СУБД MySQL 5.1 [Электронный ресурс] – URL: <http://www.mysql.com/>

14 Документация по СУБД SQLite [Электронный ресурс] – URL: <https://www.sqlite.org/>

15 Модели баз данных СУБД [Электронный ресурс] – URL: <https://www.internet-technologies.ru/articles/modeli-baz-dannyh-sistemy-upravleniya-bazami-dannyh.html>

16 Системы документооборота [Электронный ресурс] – URL: <https://www.studentlibrary.ru/doc/ISBN9785394017384-SCN0000.html>

					27.03.04.2021.799.00.00 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		85