

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
вычислительной техники

Кафедра
МОиПЭВМ

Направление подготовки 09.03.02 «Информационные системы и технологии»

Направленность Информационные системы и технологии

БАКАЛАВРСКАЯ РАБОТА

на тему

Разработка биллиноговой системы для проекта

1С: Аренда

Студент	_____	Андреев Роман Сергеевич
	(подпись, дата)	(ФИО полностью)
Руководитель	_____	Попова Н.А.
	(подпись, дата)	(фамилия, инициалы)
Нормоконтролёр	_____	Такташкин Д.В.
	(подпись, дата)	(фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от _____ № _____)

Заведующий кафедрой	_____	Козлов А.Ю.
	(подпись)	(фамилия, инициалы)

Работа защищена с отметкой _____ (протокол заседания ГЭК от _____ № _____)

Секретарь ГЭК	_____	Попова Н.А.
	(подпись)	(фамилия, инициалы)

Пенза, 2025

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

Кафедра

МО и ПЭВМ

«Утверждаю»
Заведующий кафедрой
_____ А.Ю. Козлов
«20» декабря 2024 г

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА**

1. Студент Андреев Роман Сергеевич гр. 21ВИ1 факультета ВТ
<Фамилия, имя, отчество полностью>

2. Тема работы Разработка биллинговой системы для прокта 1С: Аренда

Тема утверждена приказом ПГУ № _____ от " _____ " _____ 20 _____

3. Руководитель работы к.т.н., доцент, доцент кафедры «МО и ПЭВМ» Попова Н.А.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Назначение: поддержка деятельности предприятий, оказывающих услуги аренды серверного оборудования.

Основные функции системы:

1. ведение нормативно-справочной информации о серверах, комплектующих, договоров и клиентах;
2. расчет арендной платы;
3. вискация дополнительных оплат;
4. учёт поступления оплат от клиентов;
5. формирование аналитических отчетов.

Технология разработки: предметно-ориентированное программирование.

Язык программирования: встроенный язык программирования 1С: Предприятие.

Среда программирования: технологическая платформа 1С: Предприятие 8.3.

5. Перечень подлежащих разработке вопросов:

1. анализ предметной области и постановка задачи;
2. проектирование программных средств;
3. реализация программных средств;
4. тестирование программных средств;
5. оформление ВКР.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ предметной области и требований к системе	10%	08.04.2025 – 14.04.2025	
2	Проектирование программных средств	20%	15.04.2025 – 25.04.2025	
3	Реализация программных средств	30%	26.04.2025 – 12.05.2025	
4	Тестирование программных средств	20%	13.05.2025 – 22.05.2025	
5	Оформление пояснительной записки	20%	23.05.2025 – 03.06.2025	

Дата выдачи задания «20» декабря 2024г

Руководитель бакалаврской работы _____
(подпись, дата)

Задание к исполнению принял студент _____
(подпись, дата)

Работу к защите допустить
Декан ФВТ _____ Л.Р. Фионова
(подпись, дата)

Реферат

Пояснительная записка содержит 71 листа, 28 рисунков, 18 таблиц, 30 использованных источника и 3 приложения.

АРЕНДА СЕРВЕРОВ, ИНФОРМАЦИОННАЯ СИСТЕМА, УЧЕТ ДЕЯТЕЛЬНОСТИ, 1С: ПРЕДПРИЯТИЕ 8.3, КОНФИГУРАЦИЯ, UML, IDEF0, IDEF1X.

Предмет работы: разработка информационной системы поддержки компании оказывающая услуги аренды серверов.

Объект работы: автоматизация деятельности предприятий оказывающее услуги аренды серверов.

Цель работы: разработка инфорационной системы поддержки деятельности предприятий, оказывающих услуги аренды серверов.

Технология разработки: платформа 1С: Предприятие 8.3.

Результаты работы: спроектирована и разработана конфигурация на платформе 1С: Предприятие 8.3 для ведения учета деятельности предприятия оказывающее услуги аренды серверов.

Область применения: предприятия, предоставляющие услуги по аренде серверов.

					ПГУ 09.03.02 – 08ВР211.4 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	Разработка биллинговой системы для проекта 1С: Аренда	Лит.	Лист	Листов
Разраб.	Андреев Р.С.						4	71
Провер.	Попова Н.А.					группа 21ВИ1		
Реценз.								
Н. Контр.	Такташкин Д.В.							
Утверд.	Козлов А.Ю.							

Содержание

Введение.....	6
1 Постановка задачи и анализ аналогов разрабатываемой системы	7
1.1 Постановка задачи	7
1.2 Анализ информационных систем аналогичного назначения	8
1.3 Выбор технологий проектирования и разработки.....	12
2 Проектирование информационной системы	13
2.1 Анализ бизнес-процессов предприятия.....	13
2.2 Проектирование логической модели базы данных	16
2.3 Проектирование программного обеспечения	18
2.3.1 Анализ прецедентов.....	18
2.3.2 Диаграмма состояний	20
2.3.3 Диаграмма деятельности.....	23
2.3.4 Диаграмма последовательности	24
2.3.5 Диаграмма кооперации.....	25
3 Планирование и расчет бюджета разрабатываемой системы.....	26
4 Реализация и тестирование системы.....	31
4.1 Реализация структур программного средства.....	31
4.2 Реализация кода информационной системы	40
4.3 Тестирование информационной системы.....	42
4.3.1 Функциональное тестирование	42
4.3.2 Тестирование чёрного ящика	44
4.3.2 Метрика Холстеда.....	50
Заключение	54
Список использованных источников	55
Приложение А (обязательное) Листинг программы	58
Приложение Б (рекомендуемое) Диаграмма компонентов	65
Приложение В (рекомендуемое) Результаты тестирования	67

Введение

Значение биллинговой системы для компаний оказывающее услуги аренды серверного оборудования важно так как на этом рынке высокая конкуренция ввиду высокого спроса на серверное оборудование. Владельцы таких компаний стремятся повысить конкурентоспособность своей компании путем введения новых и современных инструментов управления. Регулярные поступления платежей от клиентов требуют регламентного учёта, правильного выставления счетов и отслеживания задолженностей или переплат.

Целью данной выпускной работы является разработка биллинговой системы для проекта 1С: Аренда, обеспечивающей эффективное управление платежами и взаимодействие с клиентами.

Для достижения этой цели были сформулированы следующие задачи:

- постановка задачи на проектирование биллинговой системы для платформы «1С: Аренда»;
- проведение анализа аналогичных систем управления по аренде серверного оборудования;
- определение главных функциональных и технических требований;
- выполнение функционального моделирования всех будущих процессов и модулей системы;
- проектирование структуры базы данных и алгоритмов обработки информации;
- реализация всех необходимых элементов системы в программной среде «1С: Предприятие 8.3»;
- тестирование готовой конфигурации для того, чтобы убедиться в работоспособности всех объектов конфигурации в соответствии со структурой.

Исходя из выполненного анализа была спроектирована конфигурация в задачи которой входит автоматизация самых важных аспектов работы компании, которая оказывает услуги аренды серверного оборудования.

1 Постановка задачи и анализ аналогов разрабатываемой системы

1.1 Постановка задачи

Сегодня трудно представить какую-либо отрасль, где бы не использовались специализированные программы и системы автоматизации. Особое внимание уделяется развитию ИТ-решений в бизнесе аренды недвижимости, поскольку качественное управление доходами и взаимодействия с клиентами становится критическим условием успеха.

Всего различают два типа информационных систем, которые применяются в управление арендным бизнесом:

Фронт-офисные системы – это системы, которые отвечают за работу с клиентами или заказчиками и являются лишь помощниками в работе компании.

Бэк-офисные системы – это системы, которые отвечают за внутреннюю работу компании и помогают управлять хозяйственными и бизнес процессами путём автоматизации тех или иных аспектов. Такие системы так же делятся на несколько подсистем:

- системы управленческого учета – применяются для оперативного контроля разных отделов компании или всей компании в целом;
- бухгалтерские и налоговые системы – применяются для систематизации всей финансовой и юридической отчетности компании.

Система которая разрабатывается в данной ВКР должна иметь несколько основных функций:

- хранение информации об оборудовании, договоров, клиентов и серверах компании;
- механизм расчёта арендной платы;
- учет дополнительных услуг;
- учет поступлений оплаты от клиентов и контроль отставков на плату от клиентов;

- формирование печатных листов договоров и актов оплаты;
- формирование аналитических отчетов.

Такое решение даст только тот функционал, который необходим предприятию без переплаты за лишний функционал, а также из-за более простой в основе системы пользователям будет проще адаптироваться к новой среде. К тому же более простое решение быстрее интегрируется в компанию и тем самым экономит много денег компании.

1.2 Анализ информационных систем аналогичного назначения

Прежде чем начинать разработку нужно произвести анализ систем-аналогов чтобы определить все достоинства и недостатки уже существующих систем для аренды серверов.

Уже существует небольшое количество систем для услуг аренды серверов, они различаются по функциональным и техническим особенностям.

Инструмент CloudHealth Technologies – является популярным инструментом для оптимизации затрат и контроля расходов в использования облачных ресурсов.

Данный инструмент работает по системе подписок, которые различаются по опциям, которые они открывают и ценой. Всего есть 3 подписки.

Free Edition: Базовая версия доступна бесплатно, но ограничивает функциональность и подходит только для начального знакомства с платформой.

Standard Edition: Предназначена для малых и средних предприятий, предлагает основные функциональные модули аналитики и рекомендаций по расходам. Цена от 500 до 1500 долларов США в месяц.

Pro Edition: Полноценная версия для крупного бизнеса, предлагающая расширенный набор возможностей, включая продвинутые отчёты, детализированные аналитические дашборды и автоматизацию действий. Цена от 2000 до 5000 долларов США в месяц.

Следующий инструмент это VMware vRealize Operations Manager (vROps). Комплексное решение для мониторинга, оптимизации и управления производительностью виртуализированных и облачных сред. Продукт разработан компанией VMware и входит в семейство инструментов vRealize, предназначенных для комплексного управления инфраструктурой и приложениями.

VMware имеет следующие тарифы.

vRealize Operations Essentials – минимальная версия, предназначенная для мониторинга и базовой аналитики производительности виртуальных машин. Подходит для небольшого парка VMware ESXi-серверов и простых виртуализированных сред. Цена от 1500 до 3000 долларов США в год за лицензию на одну виртуальную машину;

vRealize Operations Advanced – средняя версия, включает все возможности Essential плюс углубленную аналитику, прогнозы производительности, оптимизацию ресурсов и рекомендациями по улучшениям. Цена от 3000 до 6000 долларов США в год за лицензию на одну виртуальную машину;

vRealize Operations Enterprise – максимальная версия, включающая полный функционал Advanced плюс расширенную аналитику затрат, автоматизацию действий, повышенную безопасность и соответствие нормам регулирования. Цена от 6000 до 12000 долларов США в год за лицензию на одну виртуальную машину.

Следующий это Arptio. Американская компания, специализирующаяся на предоставлении программного обеспечения для управления информационными технологиями (ITFM – Information Technology Financial Management). Платформы Arptio предназначены для анализа затрат, оптимизации IT-инфраструктуры и принятия обоснованных управленческих решений относительно инвестиций в технологии.

Arptio предлагает два пакета, различающихся по функциональности и стоимости.

Essential Pack (Базовый пакет) включает основной функционал для отслеживания и анализа затрат, оптимизацию расходов и составления планов. Данный пакет подойдет малым и средним предприятиям, которые хотят внедрить систему управления затратами без сложных технических деталей. Цена от 500 до 5000 долларов США в месяц.

Enterprise Pack (Полный пакет) добавляет расширенные возможности, такие как прогнозирование расходов, создание отчетности высокого уровня, интеграция с существующими системами ERP и BI, расширенное планирование и поддержка аналитики больших объемов данных. Этот пакет предпочтителен для крупных компаний с развитой инфраструктурой и высокими требованиями к контролю и управлению ИТ-процессами. Цена от 5000 до 50000 долларов США в месяц.

Zabbix – это бесплатное и открытое решение для мониторинга и управления ИТ-инфраструктурой, которое доступно в двух версиях: бесплатной Community-версии и коммерческой Professional-версии.

Поскольку Zabbix выпускается в двух вариантах, цена сильно может отличаться.

Community Edition абсолютно бесплатна и доступна каждому пользователю без ограничения срока использования.

Professional Edition: коммерческий выпуск поставляется по платной схеме подписки, зависящей от количества узлов и уровня требуемого сервиса.

Ориентировочные тарифы (ценообразование меняется в зависимости от региона).

Подписка на поддержку и обновления для одного узла от \$50 до \$200 в год.

Пакет для крупных инсталляций с сотнями узлов от \$5000 до \$20000 в год.

Услуги VIP-поддержки с гарантированным временем отклика: от \$10000 в год и выше.

Рассмотрев представленные аналоги и оценив все их функции был проведён анализ, результаты которого приведены в таблице 1.

Таблица 1 – Сравнение систем аналогов

Система	Достоинства	Недостатки
CloudHealth Technologies	Гибкая система мониторинга и управления затратами. Интеграция с различными облачными платформами. Поддержка автоматизации и оптимизации ресурсов.	Сложность в освоении для новичков. Высокая стоимость лицензии. Ограниченные возможности для малых предприятий.
VMware vRealize Operations Manager	Высокая степень точности анализа затрат. Возможность сопоставления расходов с бизнес-задачами. Мощные инструменты прогнозирования и планирования.	Сложность в освоении для новичков. Высокая стоимость лицензии. Ограниченные возможности для малых предприятий.
Apptio	Полная автоматизация управления производством. Интеграция с 1С. Закрытая база данных и низкие системные требования. Высокая производительность специализированного сервера.	Высокая стоимость внедрения. Требуется наличия квалифицированных специалистов. Ограниченные возможности для малых организаций.
Zabbix	Бесплатная версия с открытым исходным кодом. Поддерживает различные платформы и устройства. Система гибкая и масштабируемая.	Требуется значительных усилий для настройки и поддержки. Ограниченные возможности для крупных предприятий. Отсутствие встроенной поддержки для некоторых функций.

Исходя из характеристик систем аналогов, проведено сравнение систем. Результат этих сравнений представлен в таблице 2. Баллы критериев обозначают следующее: 1 – низкое проявление критерия, 2 – среднее проявление критерия, 3 – высокое проявление критерия. Таблица критериев представлена под номером 2.

Таблица 2 – Критериальная оценка систем-аналогов

Система	Критерий				
	Функциональная сложность организации системы	Качество интерфейса	Анализ данных	Стоимость	Производительность
CloudHealth Technologies	3	1	2	3	3
VMware vRealize Operations Manager	2	1	2	3	3
Apptio	2	2	3	3	2
Zabbix	3	3	3	3	2

Оценки показывают, что многие современные системы управления серверами стоят дорого и имеют много функций, которые не нужны малым бизнесам. Это создает необходимость в разработке простой биллинговой системы для проекта «1С: Аренда», которая будет сосредоточена на основных потребностях небольших арендодателей и арендаторов.

Новое программное обеспечение поможет компаниям сэкономить на внедрении и обслуживании системы, так как в нем не будет лишних функций. Более простой интерфейс уменьшит нагрузку на сотрудников и поможет пользователям быстрее освоиться, что также сократит затраты на обучение. Такой подход даст хороший баланс между функциями и ценой, что поможет малым предприятиям лучше вести бизнес в сфере аренды серверного оборудования.

1.3 Выбор технологий проектирования и разработки

Разработки информационной системы для аренды серверов включает в себя создание UML – диаграмм, функциональных диаграмм с использованием IDEF0 и моделей данных по IDEF1X. Для были выбраны такие инструменты, как ERwin Process Modeler, ERwin Data Modeler 7.3, ArgoUML и Visual Paradigm.

ERwin Process Modeler – CASE-средство, которое помогает строить процессы по IDEF0 и управлять потоками данных в DFD. ERwin Data Modeler предназначен для создания логических и физических моделей данных и генерирует код для разных баз данных.

ArgoUML – это инструмент для работы с UML-диаграммами, Visual Paradigm поддерживает UML 2.0, SysML и моделирование бизнес-процессов. Кроме разработки диаграмм, он также позволяет создавать отчеты и генерировать код.

В данной работе были выбраны именно эти программы, потому что они имеют все необходимые функции, просты в использовании и предлагают бесплатные версии.

Для разработки системы используется платформа 1С: Предприятие 8.3. Это удобная среда, которое позволяет разрабатывать прикладные решения, предназначенные для автоматизации процессов в различных компаниях.

2 Проектирование информационной системы

Проектирование информационной системы представляет собой процесс разработки проектно-конструкторских решений, направленных на создание проекта ИС, соответствующего заданным требованиям. Это означает преобразование исходных сведений об объекте проектирования в детальное техническое описание будущей системы. Итоговым результатом этапа проектирования становится формирование архитектуры системы, определяющей её структуру и организацию на всех дальнейших стадиях жизненного цикла.

2.1 Анализ бизнес-процессов предприятия

Анализ процессов в организации, занимающейся арендой серверного оборудования, начинается с построения функциональной модели, которая наглядно отображает ключевые операции и их взаимодействие с внешним окружением. Стандарт IDEF0 идеально подходит для решения этой задачи благодаря своей способности ясно представлять иерархию процессов и информационные потоки.

Первым этапом становится определение основных функций, таких как работа с клиентами, управление инфраструктурой и техническое

сопровождение. Далее выявляются входные и выходные потоки, включая запросы клиентов и выданные подтверждения. Важно также учитывать внешние объекты – заказчиков услуг, поставщиков оборудования и техников поддержки.

Затем создается общая диаграмма контекста, представляющая весь спектр операций организации единым блоком. Этот этап важен, поскольку позволяет увидеть картину целиком перед углубленным изучением отдельных аспектов.

Инструменты типа ERwin Process Modeler значительно упрощают создание моделей, предоставляя удобный интерфейс для визуализации и документирования всех этапов проектирования. Таким образом, расширяя начатое исследование, мы получаем детальную схему функционирования всей системы, обеспечивающую прозрачность и согласованность будущих разработок.

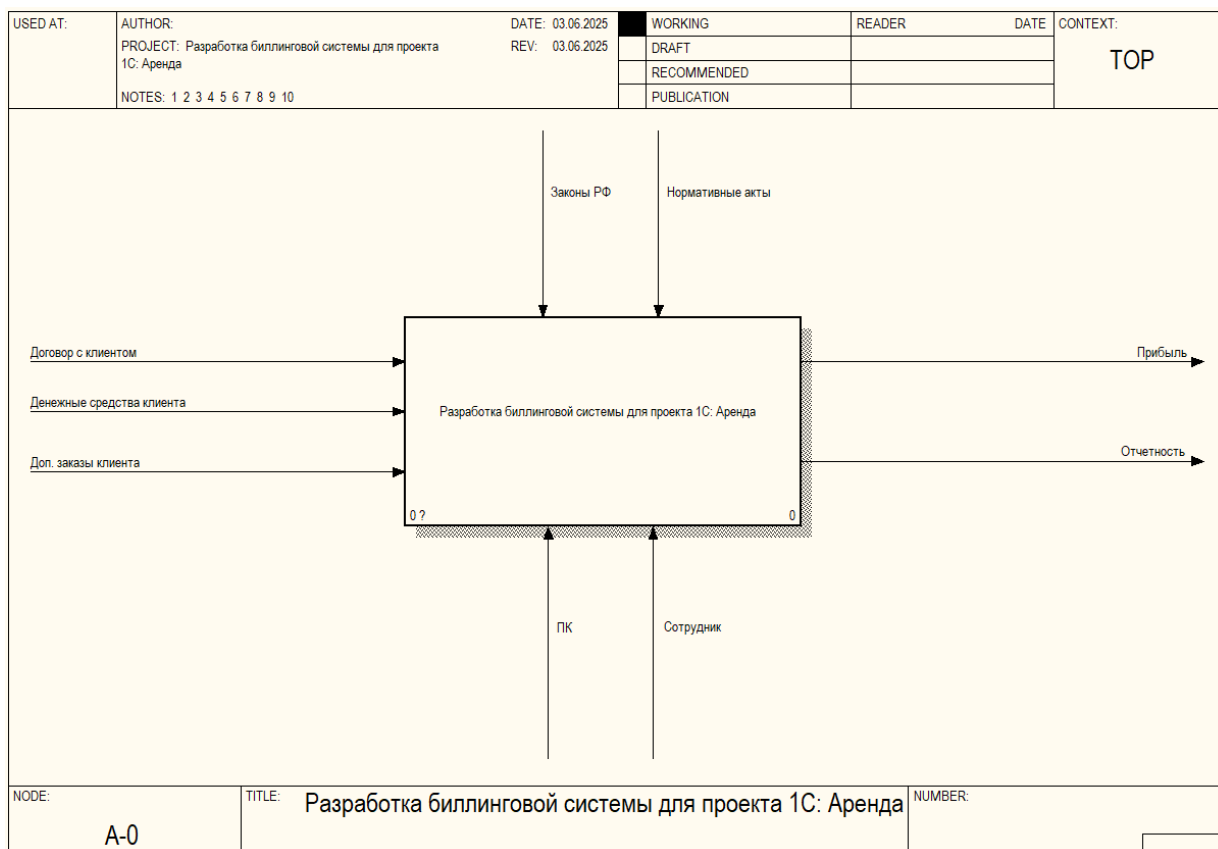


Рисунок 1 – Контекстная диаграмма

Основной блок-диаграммы был разделён на составные части, которые показывают процессы на более низком уровне. Этот процесс называется декомпозиция, результат представлен на рисунке 2.

При декомпозиции основного блока были выделены 4 процесса: «Заключение договора», «Оказание услуг по договору», «Оказание дополнительных услуг», «Анализ задолжности клиентов».

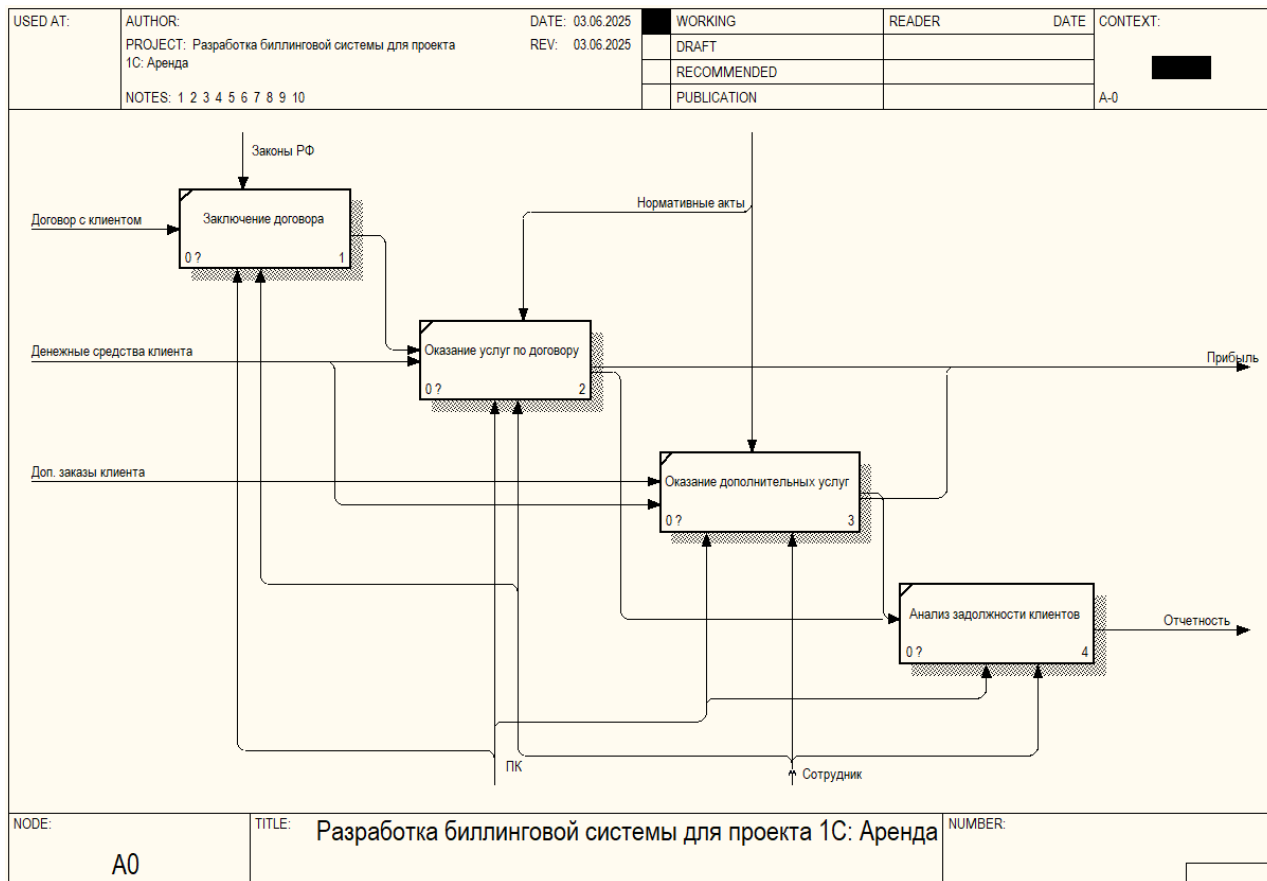


Рисунок 2 – Декомпозиционная диаграмма контекстного блока

Процесс «Заключение договора» подразумевает под собой создание договора с клиентом. Если клиента нет в базе, то сначала мы добавляем клиента в базу и только потом создаём договор, в котором указываем предмет договора. В договоре так же сразу рассчитывается стоимость оказания услуг.

Процесс «Оказание услуг по договору» подразумевает под собой указание на то, что услуги по договору были выплачены и клиенту был выставлен счёт на оплату по договору.

Процесс «Оказание дополнительных услуг» подразумевает под собой то, что клиенту были оказаны дополнительные услуги вне договора за которые были выставлены отдельные счета на оплату.

Процесс «Анализ задолжности клиентов» подразумевает под собой просмотр задолжностей или переплат от клиентов для урегулирование денежных отношений между клиентом и компанией.

2.2 Проектирование логической модели базы данных

Логическая модель типа «сущность-связь», также называемая ER-моделью (Entity–Relationship Model), представляет собой универсальный инструмент описания структуры предметной области посредством простой и понятной графической схемы. Эта модель ориентирована на удобочитаемость как разработчиками информационных систем, так и простыми пользователями, позволяя детально разобраться в структуре данных и процессах взаимодействия между объектами без привязки к физической реализации базы данных.

Основой ER-модели являются сущности, представляющие важные объекты реального мира (такие как пользователи, товары, заказчики, отделы). Эти сущности представлены графически в виде прямоугольников, каждый из которых включает набор атрибутов, характеризующих объект (ФИО сотрудников, названия товаров, адреса клиентов).

Важнейшей частью модели выступают связи между сущностями, показывающие отношения и зависимости между различными элементами системы. Такие связи бывают трёх типов: «один-к-одному», «один-ко-многим» и «многие-ко-многим». Например, связь «клиент-заказ» иллюстрирует ситуацию, когда один клиент может разместить несколько заказов, тогда как каждый отдельный заказ принадлежит одному клиенту. Логическая модель максимально приближена к готовому решению, но всё ещё относится к концептуальному этапу. Такая модель работает с понятиями сущность, атрибуты и связь. Готовая логическая модель в виде диаграммы

продемонстрирована на рисунке 3. Данна диаграмма содержит 9 сущностей, которые связаны между логическими связями.

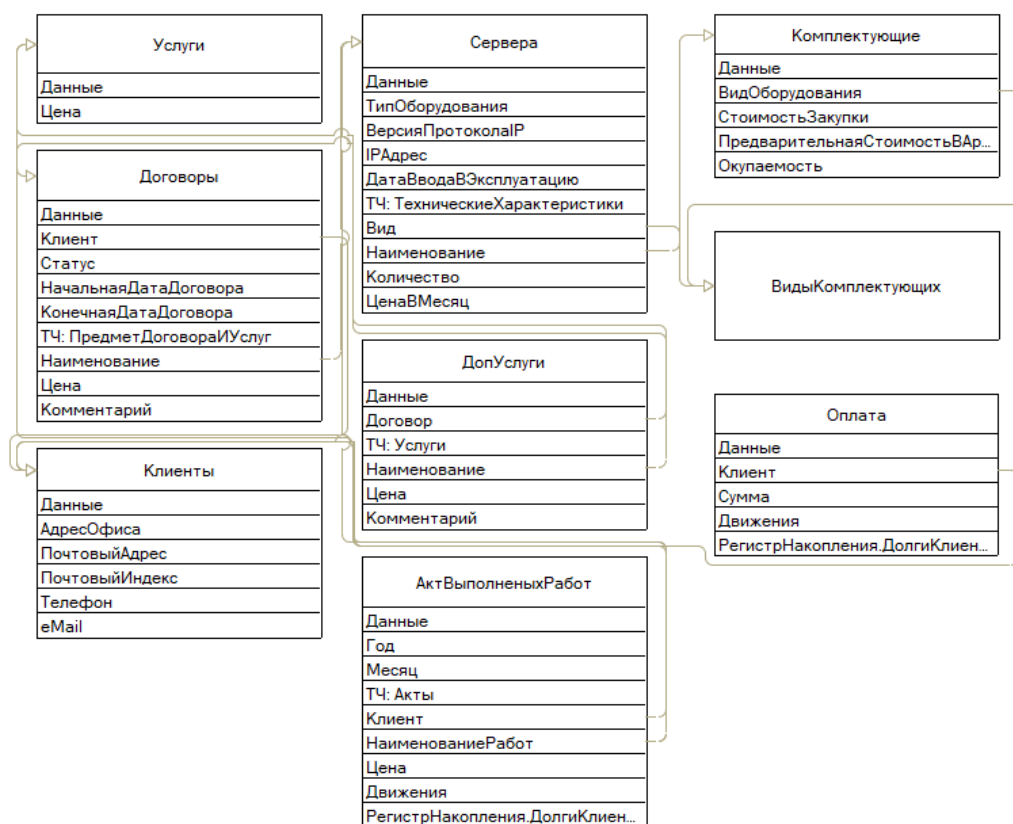


Рисунок 3 – Логическая диаграмма данных

Описание всех сущностей и атрибутов логической диаграммы приведены в таблице 3.

Таблица 3 – Описание сущностей и атрибутов логической диаграммы

Сущность	Атрибут	Роль атрибута	Описание атрибута
Услуги	Код	РК	Уникальный код организации в системе
	Код	FK	Уникальный код вида организации в системе
	Наименование организации		Название организации
	КПП		Код причины постановки на учет организации
	ОГРН		ОГРН организации
	ИНН		ИНН организации
	ОГРНИП		ОГРНИП организации
	Дата регистрации		Дата государственной регистрации

После окончания построения логической диаграммы нужно произвести нормализацию данных – исключить избыточность, протеворечивость и повысить надёжность данных, которые хранятся в системе, сделать систему, которая будет легко масштабироваться и которую легко будет адаптировать к изменениям.

Стоит так же отметить, что сущности в данной логической диаграмме имеют простые атрибуты и все не главные атрибуты зависят от первичного ключа и не зависят от остальных атрибутов. Исходя из этого следует то, что данная диаграмма и модель являются нормальной и находятся в 3 нормальной форме.

2.3 Проектирование программного обеспечения

Проектирование программного оборудования – это процесс создания проекта. Цель проектирование – определение всех свойств и требований, которые выставляются перед программным обеспечением.

Проектирование программного обеспечения включает в себя построение диаграмм, которые показывают структура программы, для этого используют язык UML который специально предназначен для описания программ.

Проектирование программного обеспечения осуществляется в программе Visual Paradigm которая является CASE-средством.

2.3.1 Анализ прецедентов

Диаграммы прецедентов (use case diagrams) в UML играют ключевую роль в процессе анализа и проектирования информационных систем. Основной целью этих диаграмм является наглядная демонстрация вариантов использования системы с точки зрения внешнего наблюдателя, чаще всего – конечного пользователя или клиента.

Графически каждый прецедент представляется эллипсом, содержащим название конкретного сценария, а акторы отображаются фигурами людей либо специальными значками, символизирующими внешний объект. Связи между

элементами показывают отношения зависимости или участия, позволяя быстро понять порядок взаимодействия и ограничения системы.

Построенная диаграмма, которая отображает актёра и его варианты действия изображена на рисунке 4.

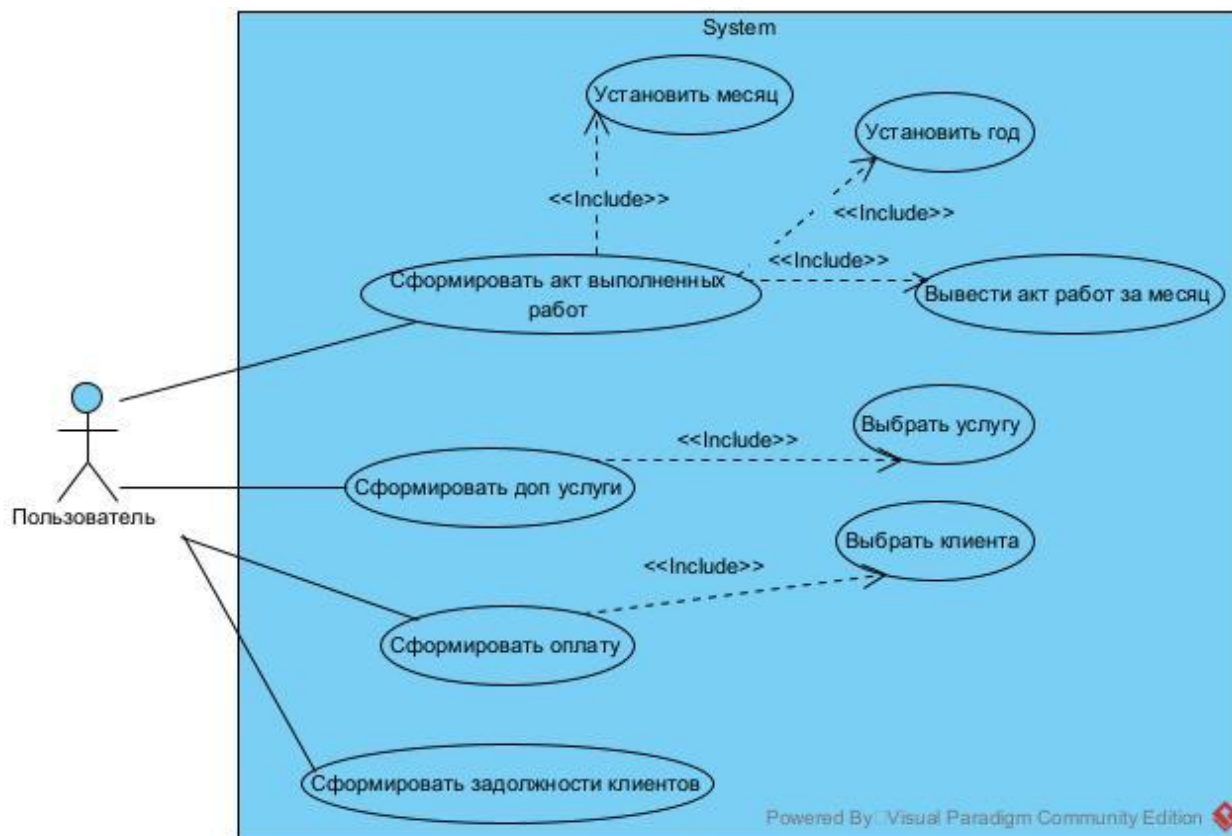


Рисунок 4 – Диаграмма вариантов использования

Описание актёров диаграммы вариантов использования и профиль их подготовки и навыки представлены в таблице 4.

Таблица 4 – Описание действующих лиц

Название	Профиль, подготовка и навыки
Пользователь	Пользователь информационной системы, который имеет большой опыт работы с программой и полный доступ к функционалу системы.

Краткое описание прецедентов (действий пользователей), действующих лиц системы и целей которые действующие лица выполняют было представлено в таблице 5.

Таблица 5 – Описание прецедентов

Действующее лицо	Цель	Краткое описание
Пользователь	Сформировать акт выполненных работ	Может создавать документ «Акт выполненных работ», который рассматривает акты выполненных работ за выбранный месяц
	Сформировать доп услуги	Может создавать документ «Доп услуги», который формирует запись услуг предоставленные организацией
	Сформировать оплату	Может создать документ «Оплата», в котором идёт выбор по контрагенту и его сумма

По результатам анализа были определены и описаны процедуры и действующие лица системы, а также построена диаграмма вариантов использования.

2.3.2 Диаграмма состояний

Диаграммы состояний являются мощным средством моделирования динамики сложных систем и позволяют детально исследовать процессы перехода объектов из одного состояния в другое. Они широко применяются в разработке программного обеспечения, анализе бизнес-процессов и проектировании технических устройств. Основная цель таких диаграмм – обеспечить четкое понимание последовательности изменений, происходящих внутри системы, и условий, приводящих к смене состояний.

Особенностью диаграммы является возможность визуализации различных сценариев развития событий, начиная от простых последовательных процессов и заканчивая комплексными ветвлениями и параллельными потоками действий. Благодаря этому обеспечивается наглядность и удобство восприятия моделируемого процесса.

Спроектирована диаграмма состояний документа «Доп услуги», представленной на рисунке 5.

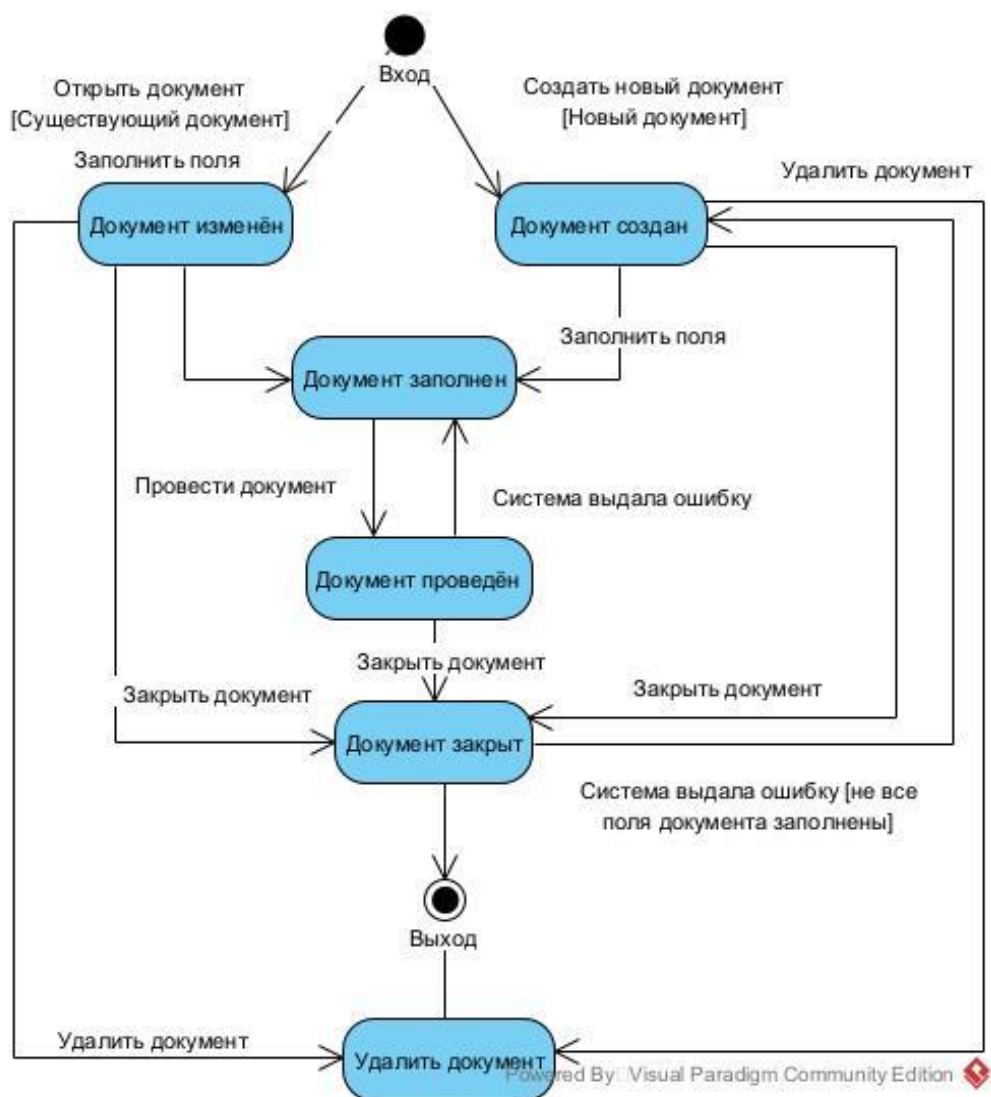


Рисунок 5 – Диаграмма состояния объекта «Доп услуги»

Спецификации переходов объекта из одного состояния в другое определены в таблице 6.

Таблица 6 – Спецификация переходов объекта из состояния в состояние

Состояние 1	Событие	Состояние 2	Описание перехода
Начальное состояние	Создать новый документ	Документ создан	При создании документ переходит из состояния начальное в состояние документ создан
Начальное состояние	Открыть документ	Документ изменён	При открытии уже существующего документа, он переходит из состояния начальное в состояние документ изменён

Продолжение таблицы 6

Состояние 1	Событие	Состояние 2	Описание перехода
Документ создан	Заполнить поля	Документ заполнен	При заполнении документа, он переходит из состояния «Документ создан» в состояние «Документ заполнен»
Документ создан	Закрыть документ	Документ закрыт	При закрытии документа, он переходит из состояния «Документ создан» в состояние «Документ закрыт»
Документ создан	Удалить документ	Документ удалён	При удалении документа, он переходит из состояния «Документ создан» в состояние «Документ удалён»
Документ изменён	Заполнить поля	Документ заполнен	При заполнении документа, он переходит из состояния «Документ изменён» в состояние «Документ заполнен»
Документ изменён	Закрыть документ	Документ закрыт	При закрытии документ, он переходит из состояния «Документ изменён» в состояние «Документ закрыт»
Документ изменён	Удалить документ	Документ удалён	При удалении документа, он переходит из состояния «Документ изменён» в состояние «Документ удалён»
Документ заполнен	Провести документ	Документ проведён	При проведении документ, он переходит из состояния «Документ заполнен» в состояние «Документ проведён»
Документ проведён	Система выдала ошибку	Документ заполнен	При проведении заполненного документ может появиться ошибка о неправильно заполненных данных в документе и перейдёт из состояния «Документ проведён» в состояние «Документ заполнен»

Таким образом, была разработана диаграмма состояний, которая показала все возможные состояния объекта «Доп. услуги».

2.3.3 Диаграмма деятельности

Диаграмма деятельности может помочь описать поток управления целевой системой, например, изучить сложные бизнес-правила и операции, описать сценарий использования, а также бизнес-процесс.

Для разработки диаграммы деятельности был выбран вариант использования «Доп услуги». Диаграмма деятельности в нотации UML приведена на рисунке 6.

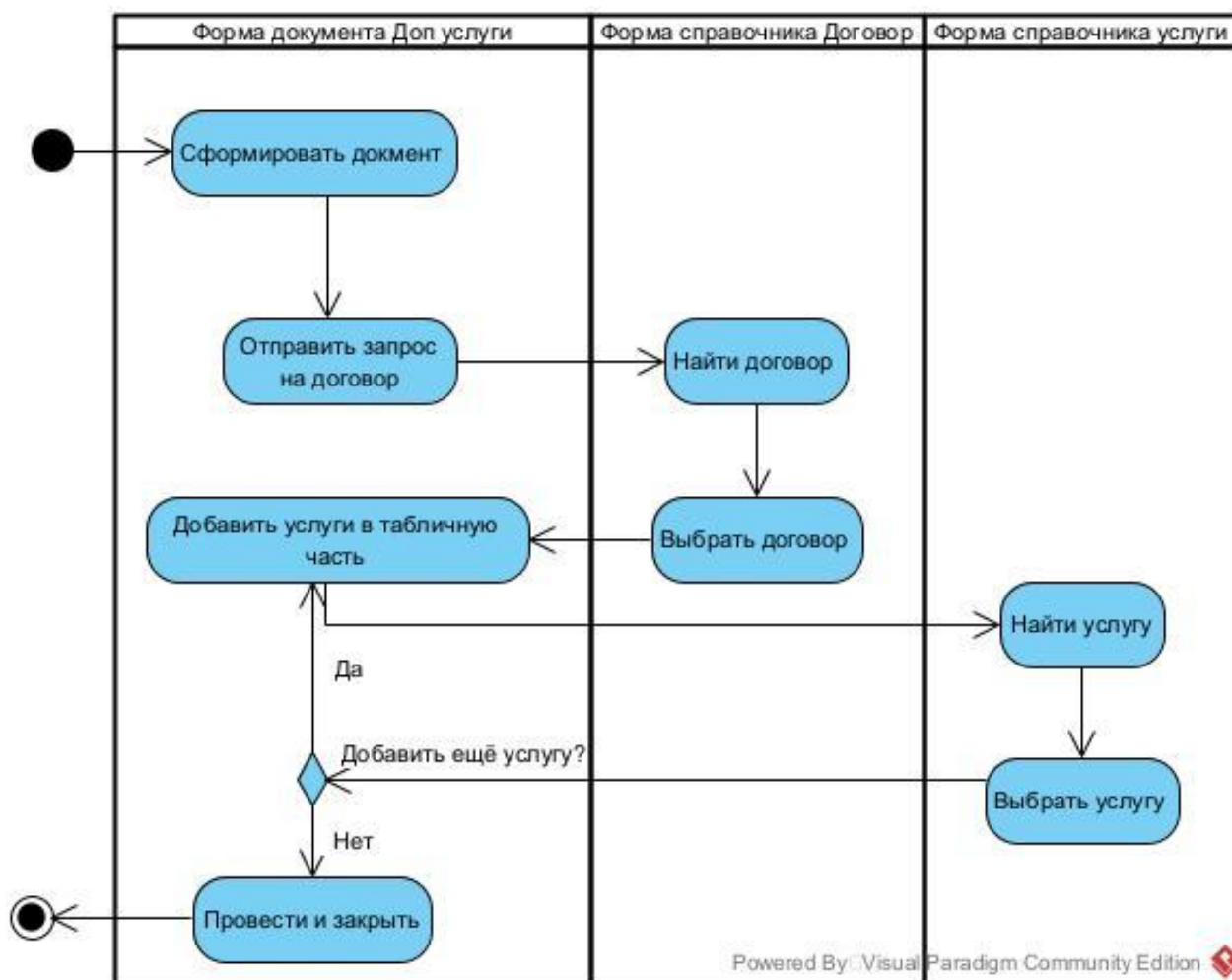


Рисунок 6 – Диаграмма деятельности

Для создания документа «Доп услуги» необходимо сформировать форму документа, то есть, нажать на кнопку «Создать» после чего открывается форма документа. В открывшейся форме необходимо установить номер

договора, на который будут указываться дополнительные услуги. При выборе документа, он выпадает в виде списка состоящий из справочника «Договор».

После этого требуется добавить услуги, которые пойдут по договору. Это действие происходит с помощью кнопки «Добавить» которая находится на форме, появляется выпадающий список из справочника «Услуги» в котором по очереди выбираем те услуги, которые войдут в договор.

Была построена диаграмма деятельности в нотации UML.

2.3.4 Диаграмма последовательности

Для визуализации взаимодействий между объектами системы применяется диаграмма последовательности.

Данная диаграмма последовательности для документа «Доп. Услуги» изображена на рисунке 7.

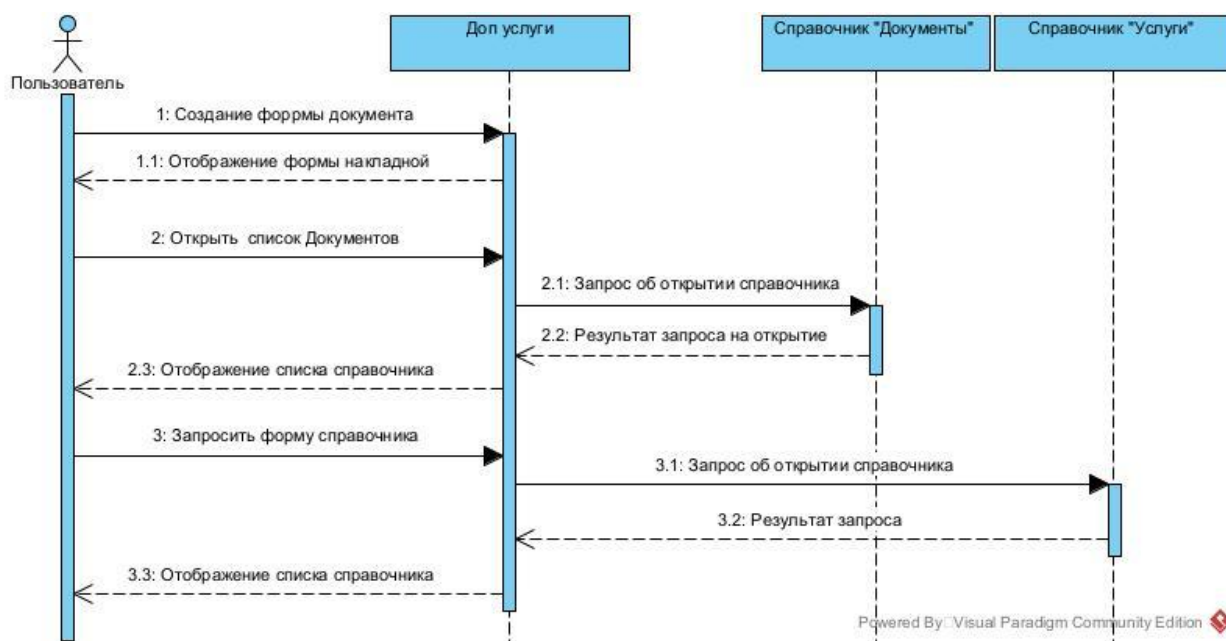


Рисунок 7 – Диаграмма последовательности

В созданной диаграмме последовательности участвует один актёр «Пользователь» и три элемента системы:

- документ «Доп услуги»;
- справочник «Документы»;
- справочник «Услуги».

Для создания документа «Доп услуги» пользователь создаёт новый документ и отображает его пользователю. После этого пользователь указывает номер документа. Чтобы его получить, необходимо запросить список документов. Запрос обрабатывается и возвращается пользователю. При выборе услуги, пользователь отправляет запрос на справочник «Услуги», система обрабатывает запрос и возвращает выпадающий список услуг.

Таким образом была построена диаграмма последовательности в нотации UML.

2.3.5 Диаграмма кооперации

Для визуализации взаимодействий и отношений между объектами в системе применяется диаграмма кооперации, которая показывает, как объекты связываются друг с другом для выполнения определённых задач.

Для разработки диаграммы кооперации был выбран документ «Доп услуги». При проектировании были выделены следующие объекты: «Пользователь», «Форма документа «Доп услуги»», «Справочник «Документы»», «Справочник «Услуги»».

Построенная диаграмма кооперации приведена на рисунке 8.



Рисунок 8 – Диаграмма кооперации

Для создания документа «Доп услуги», открывается форма документа. Пользователь раскрывает список документов. После этого идёт добавление услуг.

Таким образом, была построена диаграмма кооперации в нотации UML.

3 Планирование и расчет бюджета разрабатываемой системы

Для того чтобы внедрение новой системы было максимально быстрое и правильное, нужно следить за ресурсами проекта. Для этого используют специальные системы по управлению проектами. Благодаря этим системам получается эффективно управлять ресурсами проекта. Такие системы позволяют анализировать задачи проекта и помогают выполнять задачи в сроки.

При проектировании систем можно столкнуться с разными сложностями. Например если в разработки системы участвует несколько человек и при – это разработка достаточно длительная, то в процессе один или несколько участников разработки начать делать не то, что нужно так как могут просто потряться, а нужно чтобы все члены команды действовали сообща на всех этапах жизненного цикла и при переходах от одного этапа к другому.

Эту задачу при помощи визуального отображения в виде графика решает диаграмма Ганта. Она показывает во временных отрезках какая задача сейчас является основной и что именно требуется от того или иного участника проекта. Эта диаграмма показывает все сроки, стар и окончания задач, а также показывает задержки и другие показатели.

Самое простой и распространённой системой для построения диаграммы Ганта является Office Project. Для данной разрабатываемой системы были определены двадцать задач, ознакомиться с ними можно в таблице 7.

Таблица 7 – Задачи проекта

№	Название задачи	Длительность	Начало	Окончание	Предшественники
1	Начало проекта	0 дней	Вт 08.04.2025	Вт 08.04.2025	
2	Анализ предметной области систем аналогов	5 дней	Вт 08.04.2025	Пн 14.04.2025	1

Продолжение таблицы 7

№	Название задачи	Длительность	Начало	Окончание	Предшественники
3	Постановка задачи	1 день	Вт 08.04.2025	Ср 09.04.2025	
4	Поиск и изучение систем аналогов	3 дней	Чт 10.04.2025	Пн 14.04.2025	3
5	Выбор технологий проектирования и разработки	1 день	Пн 14.04.2025	Вт 15.04.2025	4
6	Проектирование ИС	10 дней	Вт 15.04.2025	Пт 25.04.2025	2
7	Анализ бизнес-процессов предприятия	3 дня	Вт 15.04.2025	Пт 18.04.2025	
8	Анализ требований	1 день	Пт 18.04.2025	Сб 19.04.2025	7
9	Проектирование логической модели БД	2 дней	Сб 19.04.2025	Пн 21.04.2025	
10	Проектирование ПО	4 дней	Пн 21.04.2025	Пт 25.05.2025	9
11	Реализация ИС	16 дней	Сб 26.04.2025	Пн 12.05.2025	6
12	Кодирование	12 дней	Сб 26.04.2025	Ср 07.05.2025	
13	Реализация пользовательского интерфейса	4 дней	Чт 08.05.2025	Пн 12.05.2025	12
14	Тестирование ИС	10 дней	Вт 13.05.2025	Чт 22.05.2025	11
15	Тестирование метрикой Холстеда	5 дней	Вт 13.05.2025	Сб 17.05.2025	
16	Функциональное тестирование	5 дней	Вс 18.05.2025	Чт 22.05.2025	15

Окончание таблицы 7

№	Название задачи	Длительность	Начало	Окончани	Предшественники
17	Подготовка документации	12 дней	Пт 23.05.2025	Вт 03.06.2025	14
18	Написание документации	12 день	Пт 23.05.2025	Пн 02.06.2025	
19	Оформление и печать документации	1 день	Пн 02.06.2025	Вт 03.06.2025	18
20	Конец проекта	0 дней	Вт 03.06.2025	Вт 03.06.2025	17

Так же были определены необходимые ресурсы. Список ресурсов представлен в таблице 8.

Таблица 8 – Ресурсы проекта

Название ресурса	Тип	Стандартная ставка	Начисление
Аналитик	Трудовой	150,00 Р/ч	Пропорциональное
Проектировщик	Трудовой	130,00 Р/ч	Пропорциональное
Программист	Трудовой	170,00 Р/ч	Пропорциональное
Тестировщик	Трудовой	150,00 Р/ч	Пропорциональное
Компьютер	Материальный	15 000,00 Р	В начале
Принтер	Материальный	2 000,00 Р	В начале
Бумага	Материальный	300,00 Р	В начале
Интернет	Затраты		В начале
Электроэнергия	Затраты		По окончании

Выделенные задачи и ресурсы были занесены в созданный проект в Office Project. Результаты представлены на рисунке 9.

	Режим задачи	Название задачи	Длительность	Начало	Окончание	Предшественник	Название ресурса	Затраты	% завершения	Добавить
1	✓	Начало проекта	0 дней	Вт 08.04.25	Вт 08.04.25			0,00 Р	100%	
2	✓	Анализ предметной области и систем аналогов	5 дней	Вт 08.04.25	Пн 14.04.25			6 000,00 Р	100%	
3	✓	Постановка задачи	1 день	Вт 08.04.25	Вт 08.04.25		Аналитик	1 200,00 Р	100%	
4	✓	Поиск и изучение систем аналогов	3 дней	Чт 10.04.25	Пн 14.04.25	2	Аналитик	3 600,00 Р	100%	
5	✓	Выбор технологий проектирования и разработки	1 день	Пн 14.04.25	Пн 14.04.25	3	Аналитик	1 200,00 Р	100%	
6	✓	Проектирование ИС	10 дней	Вт 15.04.25	Пн 28.04.25	1		10 000,00 Р	100%	
7	✓	Анализ бизнес-процессов	3 дней	Вт 15.04.25	Чт 17.04.25		Проектировщик	3 120,00 Р	100%	
8	✓	Анализ требований	1 день	Пт 18.04.25	Пт 18.04.25	6	Проектировщик	1 040,00 Р	100%	
9	✓	Проектирование логической модели	2 дней	Сб 19.04.25	Вт 22.04.25	7	Проектировщик	2 080,00 Р	100%	
10	✓	Проектирование ПО	4 дней	Пн 21.04.25	Чт 24.04.25	8	Проектировщик	4 160,00 Р	100%	
11	✓	Реализация ИС	16 дней	Сб 26.04.25	Пн 19.05.25	5		21 760,00 Р	100%	
12	✓	Кодирование	12 дней	Сб 26.04.25	Вт 13.05.25		Программист	16 320,00 Р	100%	
13	✓	Реализация пользовательского интерфейса	4 дня	Чт 08.05.25	Вт 13.05.25	11	Программист	5 440,00 Р	100%	
14	✓	Тестирование ИС	4 дня	Вт 13.05.25	Пт 16.05.25	10		4 800,00 Р	100%	
15	✓	Тестирование метрик	2 дня	Вт 13.05.25	Ср 14.05.25		Тестирующий	2 400,00 Р	100%	
16	✓	Функциональное тестирование	2 дня	Вс 18.05.25	Вт 20.05.25	14	Тестирующий	2 400,00 Р	100%	
17	✓	Подготовка документации	2 дня	Пт 23.05.25	Пн 26.05.25	13		2 400,00 Р	100%	
18	✓	Написание документации	1 день	Пт 23.05.25	Пт 23.05.25		Аналитик	1 200,00 Р	100%	
19	✓	Оформление и печать документации	1 день	Пн 02.06.25	Пн 02.06.25	17	Аналитик	1 200,00 Р	50%	
20	✓	Конец проекта	0 дней	Вт 03.06.25	Вт 03.06.25	16		0,00 Р	0%	

Рисунок 9 – Задачи проекта с распределёнными по ним ресурсами в Microsoft Office Project

В соответствии с занесёнными задачами и ресурсами была построена диаграмма Ганта, представленная на рисунке 10.

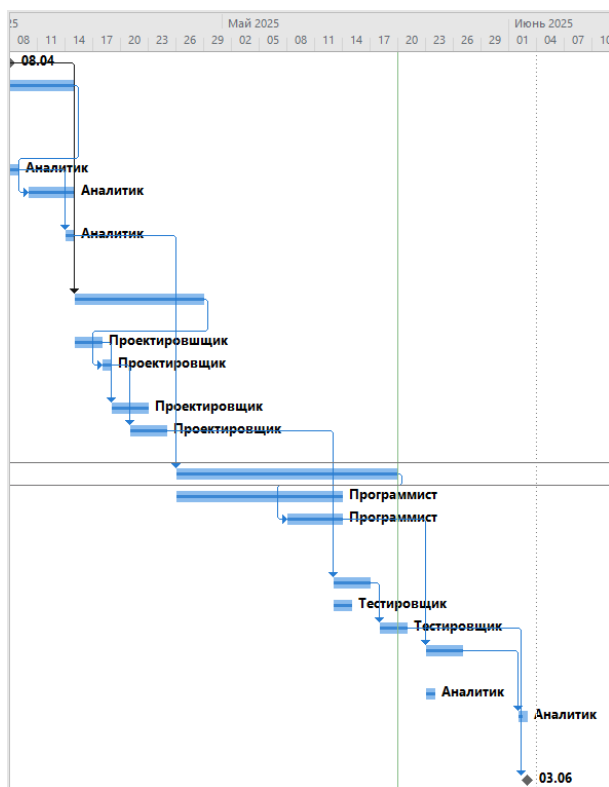


Рисунок 10 – Диаграмма Ганта

В соответствии с построенной диаграммой был сформирован отчёт по затратам на разработку, приведённый на рисунке 11.

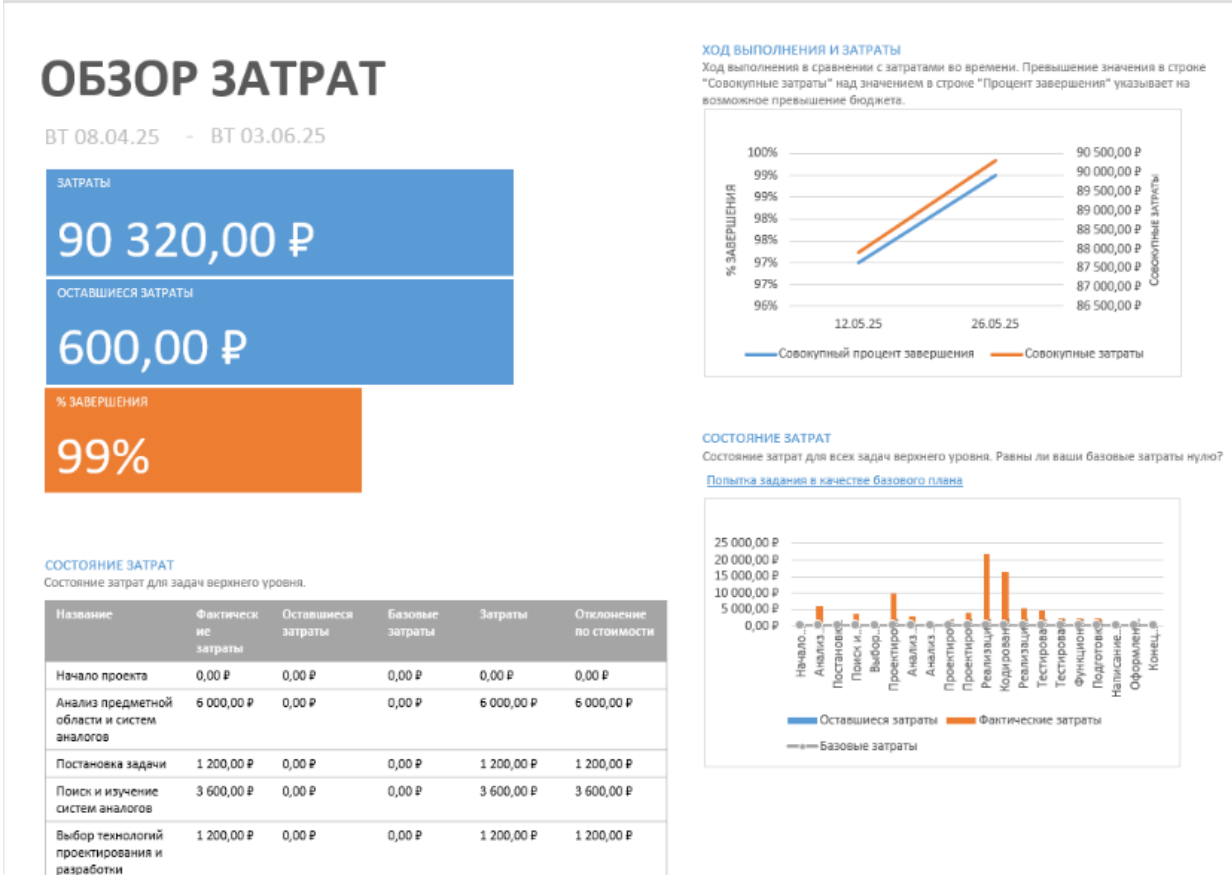


Рисунок 11 – Обзор затрат

Таким образом, по результатам построения диаграммы Ганта видно, что срок разработки проекта составил 46 дней включая выходные, бюджет проекта составляет 90 320 рублей.

4 Реализация и тестирование системы

4.1 Реализация структур программного средства

При реализации системы на платформе «1С: Предприятие 8.3» были созданы объекты, которые отображают позволяют отобразить необходимые бизнес – процессы, которые были описаны выше. Визуальное отображение созданных объектов отображено на рисунке 12, на котором, изображена диаграмма конмпонентов.

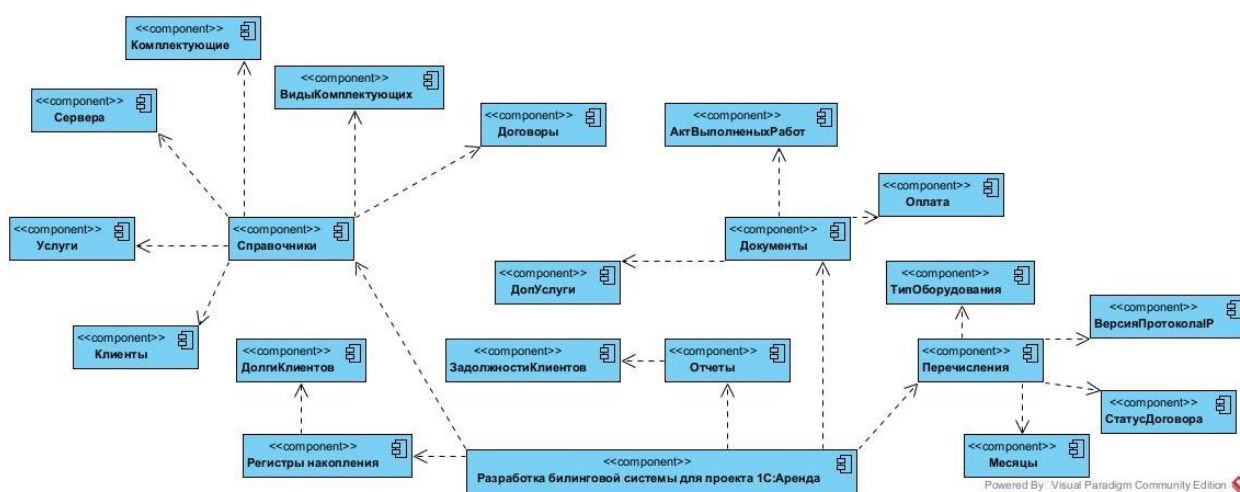


Рисунок 12 – Диаграмма компонентов

Для визуального и практического удобства пользователей все объекты находятся в подсистемах, что позволило логически скомпоновать все объекты вместе, подсистемы представлены на рисунке 13.

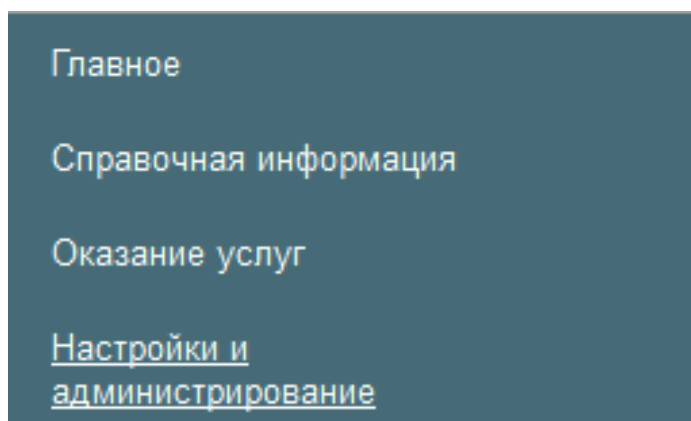


Рисунок 13 – Подсистемы

В каждой подсистеме есть подчинённые подсистемы, где и находятся объекты системы. Подсистемы позволили более корректно организовать компоновку объектов, подчинённые подсистемы представлены на рисунке 14.

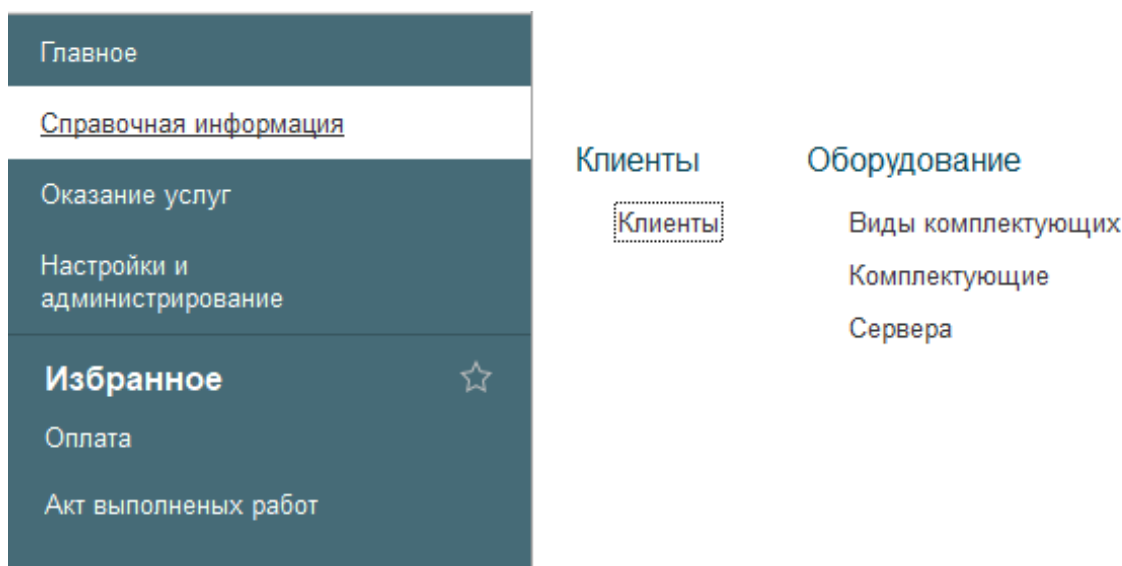


Рисунок 14 – Подчинённые подсистемы

Справочник «Клиенты» необходим для хранения всех клиентов организации. В нём хранится такая информация как ФИО клиента, его номер телефона, почта, email и почтовый индекс, справочник «Клиенты» представлен на рисунке 15.

Наименование	Код	Адрес офиса	Почтовый адрес	Почтовый индекс	Телефон	e-mail
ИПТ "Пенза"	000000004	г. Пенза ул. Погова 72	г. Пенза ул. Погова 72	440046	+7 (869) 864 32 23	ivanov58@gmail.com
ИПТ Агасян	000000002	ул. Мира 34	ул. Мира 34 к/а 1	434545	+7 (865) 665 45 45	aga@gmail.com
ООО "101"	000000001	ул. Московская 101	ул. Московская 101 офис 101	440044	+7 (999) 273 43 13	101@gmail.com
ООО "Стар01"	000000003	г. Пенза ул. Луначарского 52	г. Пенза ул. Луначарского 52	560007	+7 (868) 997 98 78	stan@gmail.com

Рисунок 15 – Справочник «Клиенты»

Справочник «Договоры» в котором хранятся все договоры, которые организация заключила с клиентами. В этом справочнике хранится дата заключения и дата окончания договора. А также ФИО клиента и предмет договора, справочник «Договоры» поедставлен на рисунке 16.

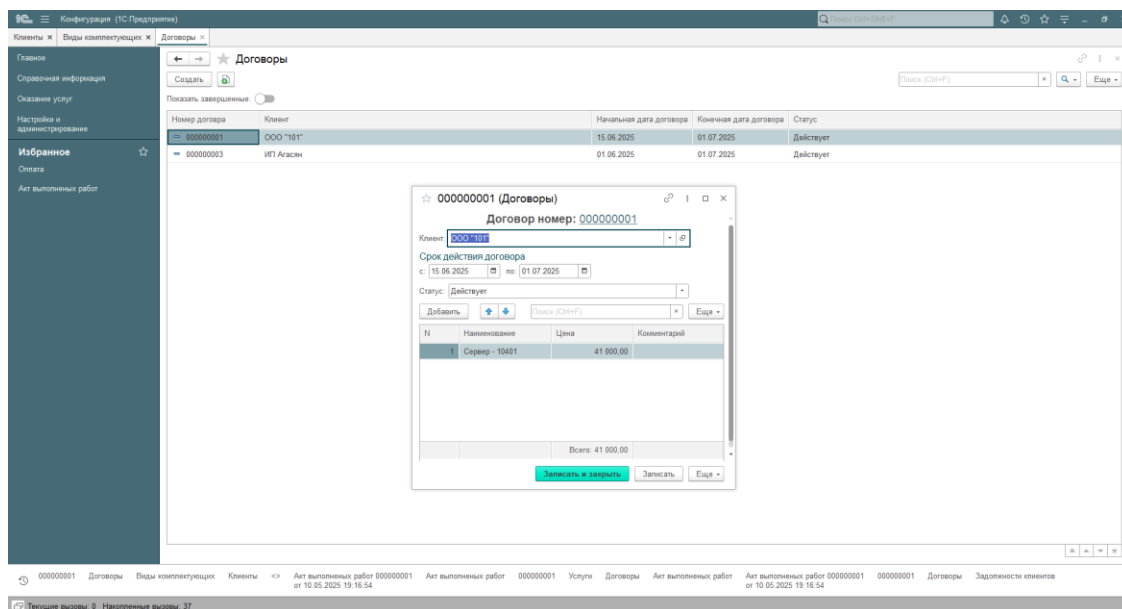


Рисунок 16 – Справочник «Договоры»

Справочник «Сервера» хранит в себе всю необходимую информацию и серверном оборудовании которая есть в организации, справочник «Сервера» представлен на рисунке 17.

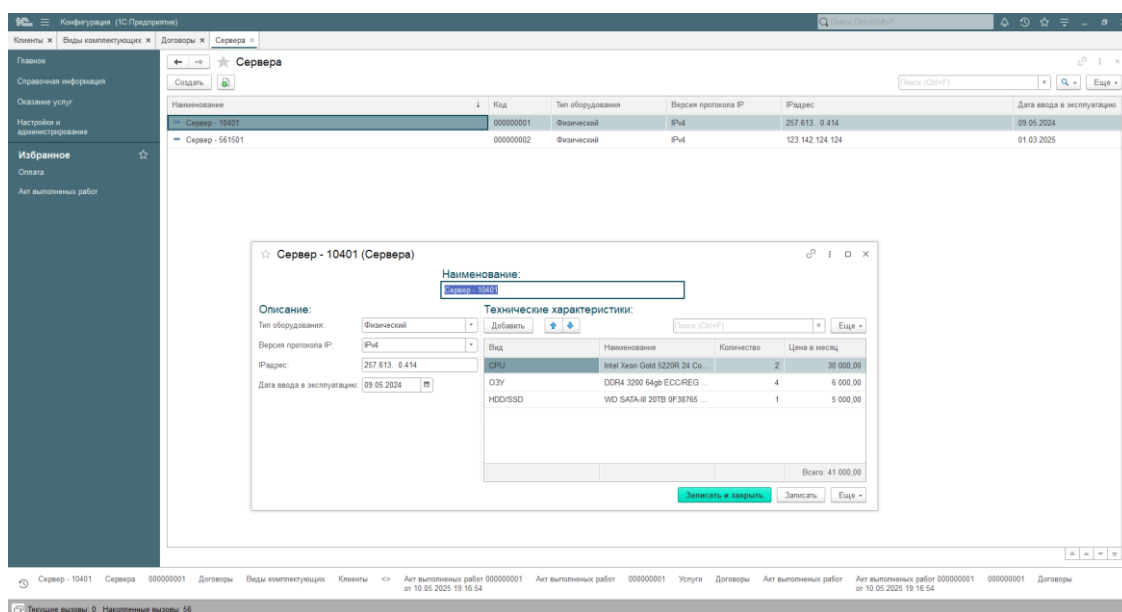


Рисунок 17 – Справочник «Сервера»

Справочник «Комплектующие» позволяет хранить перечень всех комплектующих кототые могут быть в серверном оборудовании, справочник «Комплектующие» представлен на рисунке 18.

The screenshot shows the 'Комплектующие' (Components) window. It features a sidebar on the left with navigation options like 'Главное', 'Справочная информация', 'Оказание услуг', 'Настройка и администрирование', 'Избранное', 'Оплата', and 'Акты выполненных работ'. The main area displays a table with columns: 'Наименование', 'Код', 'Вид оборудования', 'Стоимость закупки', 'Предварительная стоимость в аренде', and 'Окупаемость'. The table lists various hardware components such as Intel Xeon processors, AMD Ryzen CPUs, NVIDIA GPUs, and different types of SSDs and RAM modules, each with a unique code and associated costs.

Наименование	Код	Вид оборудования	Стоимость закупки	Предварительная стоимость в аренде	Окупаемость
Корпус ITX PCie 100W 4.0 x4 18.2 2290 500000	000000006	HDD/SSD	4 900.00	700.00	7
ADATA XPG GAMMIX D29 [AX4U3200320 16A-DCBK20] 64 GB	000000005	ОЗУ	10 500.00	1 500.00	7
ASUS GeForce RTX 4090 TUF Gaming OG OC Edition [TUF-RTX4090-O24G-OG-GAMING]	000000008	GPU	270 000.00	60 000.00	5
DDR4 3200 64Gb ECC REG Samsung CL22 M393A8G4AB2-CWEEO	000000003	ОЗУ	13 500.00	1 500.00	9
i9 - 10900K	000000001	CPU	36 000.00	5 000.00	7
Intel Xeon Gold 6220R 24 Cores, 48 Threads, 2.2/4.0GHz	000000002	CPU	135 000.00	15 000.00	9
MSI GeForce RTX 4090 GAMING X SLIM [GeForce RTX 4090 GAMING X SLIM 24G]	000000007	GPU	250 000.00	50 000.00	5
WD SATA-III 20TB 8738765 VLUH72200BL6E64 Server Ultrastar DC HC560 512E (7200rpm)	000000004	HDD/SSD	48 000.00	5 000.00	10

Рисунок 18 – Справочник «Комплектующие»

Справочник «Виды комплектующих» хранит все виды комплектующих серверного оборудования и позволяет выбирать эти комплектующие с отбором по необходимому типу, справочник «Виды комплектующих» представлен на рисунке 19.

The screenshot shows the 'Виды комплектующих' (Types of components) window. The sidebar is identical to the previous screenshot. The main area displays a table with columns: 'Наименование' and 'Код'. The table lists general categories of components: CPU, GPU, HDD/SSD, and OZU (Memory), each with a unique code.

Наименование	Код
CPU	000000001
GPU	000000002
HDD/SSD	000000004
ОЗУ	000000003

Рисунок 19 – Справочник «Виды комплектующих»

Справочник «Услуги» хранит перечень всех дополнительных услуг, которые организация может оказывать клиентам, справочник «Услуги» представлен на рисунке 20.

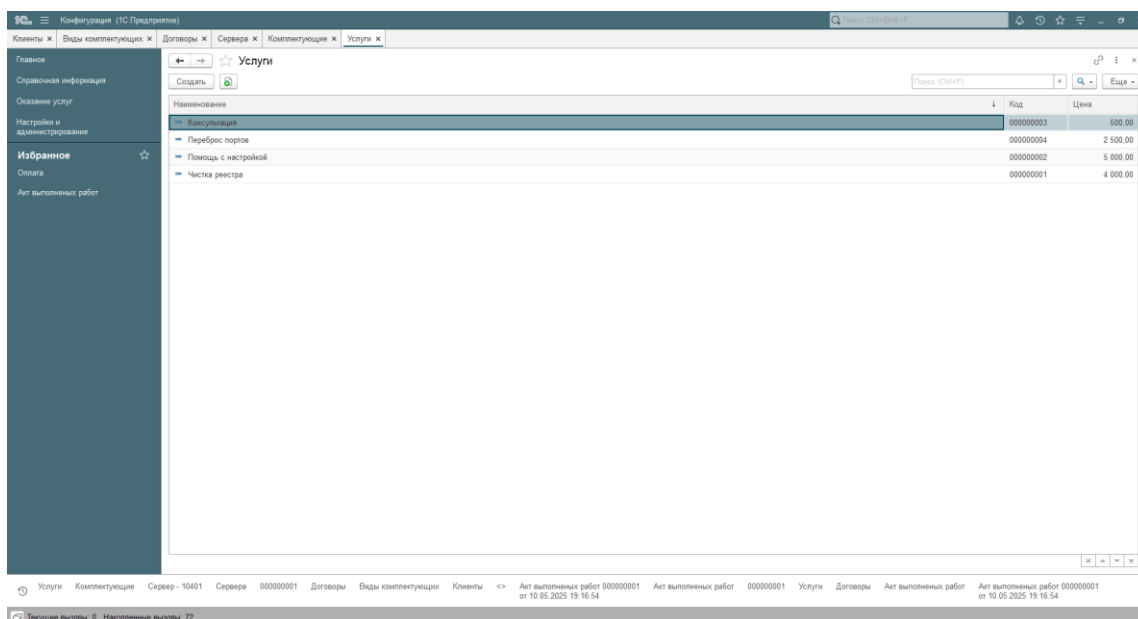


Рисунок 20 – Справочник «Услуги»

Документ «Акт выполненных работ» позволяет проводить и сохранять операции, которые показывают выполнение работ по договору с клиентом и выставлять счёт за эти работы, документ «Акт выполненных работ» представлен на рисунке 21.

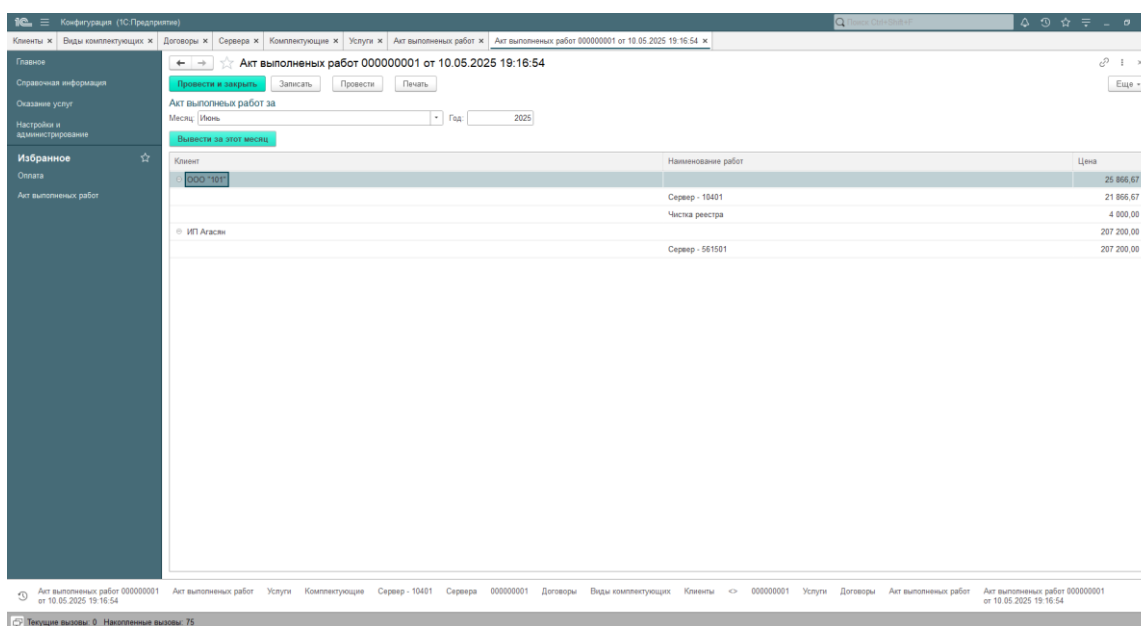


Рисунок 21 – Документ «Акт выполненных работ»

Документ «Доп. услуги» позволяет сохранять операции по выполнению дополнительных услуг клиенту и выставлять за это счёт вместе со счетом по договору, документ «Доп. услуги» представлен на рисунке 22.

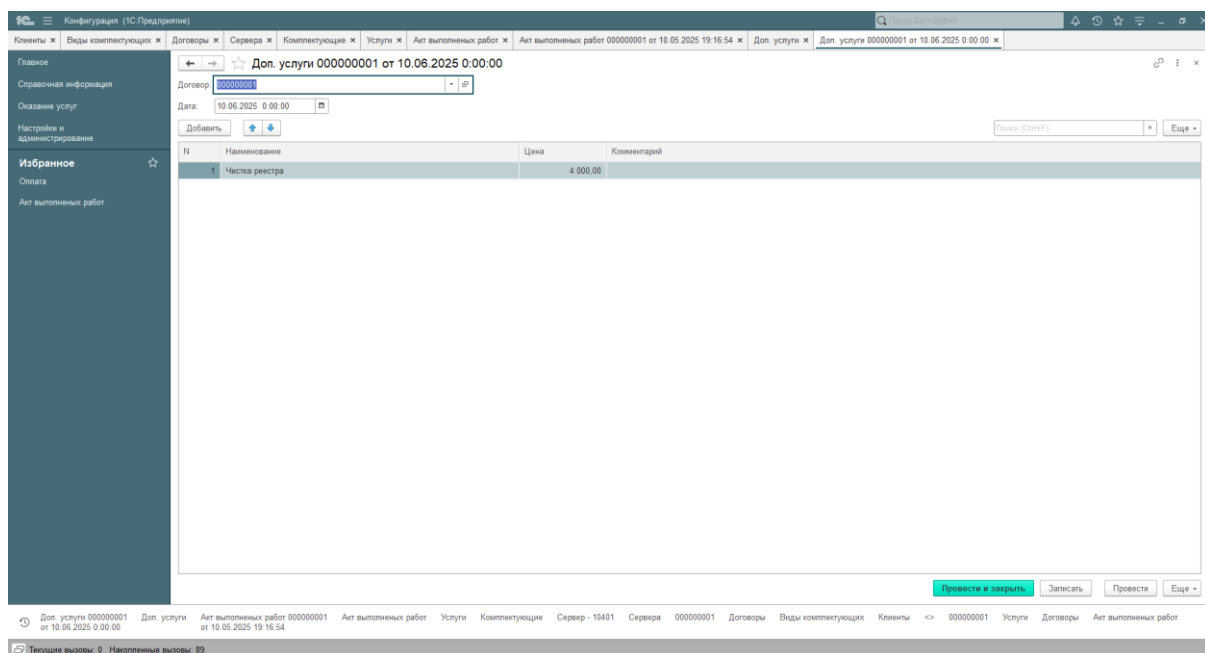


Рисунок 22 – Документ «Доп услуги»

Документ «Оплата» подтверждает оплату клиента по выставленному счету, документ «Оплата» представлен на рисунке 23.

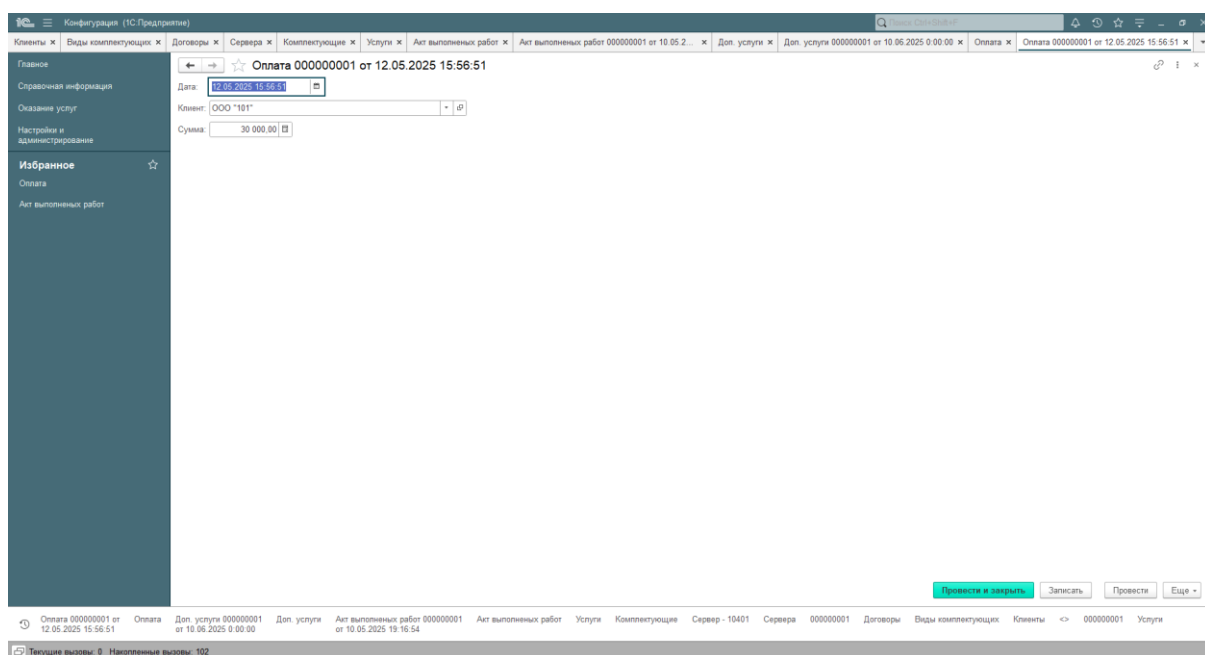


Рисунок 23 – Документ «Оплата»

У документа «Акт выполненных работ» есть печатная форма, которая позволяет передать информацию об оплате клиенту, печатная форма документа «Акт выполненных работ» предстала на рисунке 24.

Акт выполненных работ за Июнь 2025 года

Заказчик: ИТ Ассист
Исполнитель: ООО "Аренда 1С"

№	Наименование работ	Цена
Сервер - 561501		207 200.00
Итого:		207 200

Всего оказано услуг 1 на сумму 207 200 руб.
Двести семь тысяч двести рублей 00 копеек.

Выше перечисленные услуги выполнены полностью и в срок. Заказчик претензий к объёму, качеству и срокам оказания услуг не имеет.

Исполнитель/подпись: _____ Заказчик/подпись: _____

Акт выполненных работ за Июнь 2025 года

Заказчик: ООО "ИТ"
Исполнитель: ООО "Аренда 1С"

№	Наименование работ	Цена
Сервер - 10401		41 000.00
Чистка реестра		4 000.00
Итого:		45 000

Всего оказано услуг 2 на сумму 45 000 руб.
Сорок пять тысяч рублей 00 копеек.

Выше перечисленные услуги выполнены полностью и в срок. Заказчик претензий к объёму, качеству и срокам оказания услуг не имеет.

Исполнитель/подпись: _____ Заказчик/подпись: _____

Рисунок 24 – Форма печати документа «Акт выполненных работ»

Так же для работоспособности системы были созданы такие объекты как перечисления. Перечисления в платформе «1С: Предприятие 8.3» представляют собой особый вид метаданных, предназначенных для хранения фиксированных наборов значений. Эти наборы задаются разработчиками один раз и остаются неизменными для пользователей, выступая как своего рода константы.

Их основное назначение заключается в ограничении выбора допустимых значений для различных элементов данных и процессах системы. Благодаря этому обеспечивается строгий контроль и унификация данных, предотвращается ввод некорректных сведений, улучшается качество обработки информации и упрощается логика приложений.

Пользователи не могут сами добавлять или изменять элементы перечислений — такая возможность доступна исключительно разработчикам конфигураций. Перечисления представлены на рисунке 25.

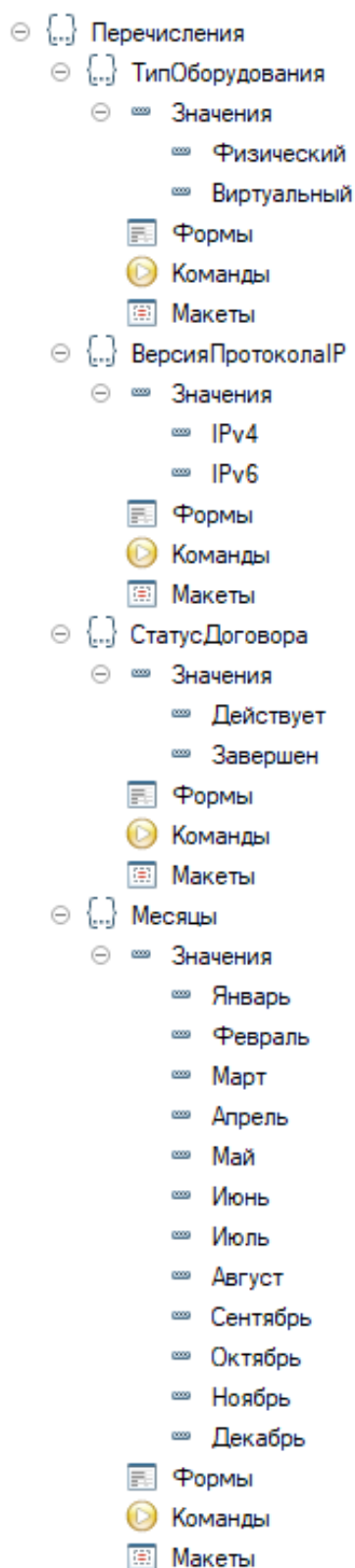


Рисунок 25 – Список перечислений в системе

В системе так же присутствует такой объект как регистр накопления «Долги клиентов» который позволяет оперативно рассчитывать есть ли

задолженность или переплата за клиентом. Регистр накоплений представлен на рисунке 26.

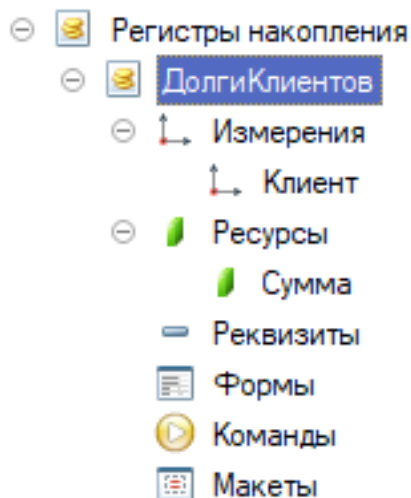


Рисунок 26 – Регистр накопления «Долги клиентов»

Для анализа задолженностей и перерасчётов клиентов в системе существует такой объект как Отчёт «Задолжности клиентов». Он позволяет визуально отобразить задолжности и перерасчёты клиентов. Отчет представлен на рисунке 27.

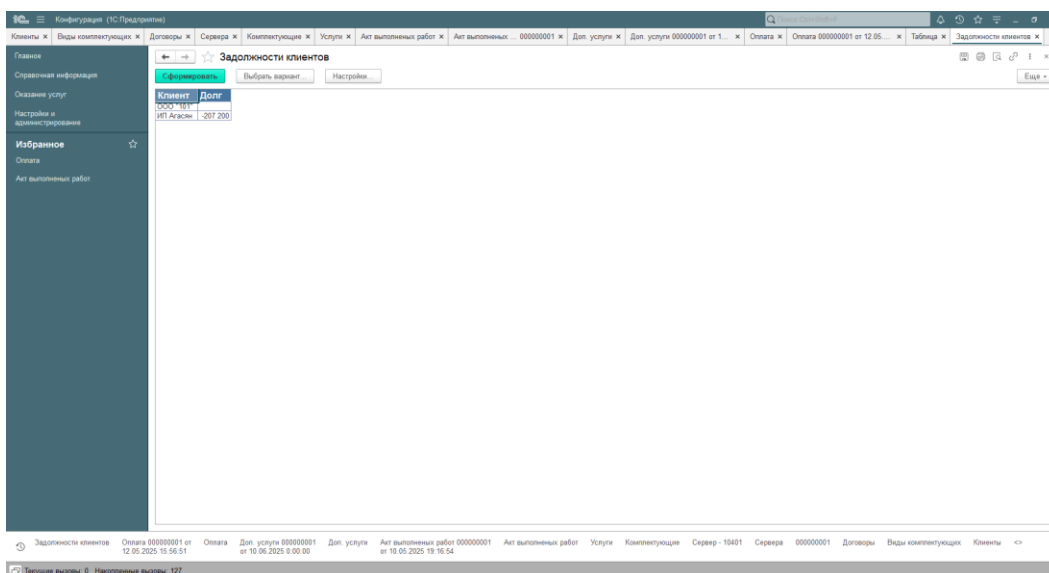


Рисунок 27 – Отчёт «Задолжности клиентов»

Из выше сказанного следует что данная система для отображения всех необходимых бизне процессов состоит из 6 справочников, 3 документов, 4 перечислений, 1 отчёта и 1 регистра накоплений.

4.2 Реализация кода информационной системы

В ходе разработки системы на платформе 1С: Предприятие 8.3 был написан код для автоматизации и работоспособности тех или иных процессов. Программный код срабатывает при тех или иных условиях, что позволяет контролировать систему при каждом взаимодействии с пользователем. Описание всех процедур и функций находится в таблице 9.

Таблица 9 – Описание всех процедур и функций системы

Объект	Функция/Процедура	Описание
Модуль формы элемента справочника «Сервера»	Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)	Процедура проверяет существование дубликата сервера в справочниках, если есть, то выдаёт ошибку
	Процедура ВерсияПротоколаIPПриИзменении(Элемент)	Процедура изменяет маску ввода кода в зависимости от IP адреса, во время изменения его протокола
	Функция ПроверитьНаименованиеНаДубликат()	Функция проверяет сервера на дубликат и возвращает ответ процедуре «Перед записью»
	Процедура ТехническиеХарактеристикиНаименованиеПриИзменении(Элемент)	Процедура срабатывает при изменении технических характеристик. Она пересчитывает цену в месяц.
	Функция ПолучитьЦенуВМесяц(Наименование)	Данная функция возвращает цену комплектующих
Модуль формы элемента справочника «Комплектующие»	Процедура СтоимостьЗакупкиПриИзменении(Элемент)	Данная процедура срабатывает при изменении стоимости закупки
	Процедура ПредварительнаяСтоимостьВАрендеПриИзменении(Элемент)	Данная процедура срабатывает при изменении Предварительной стоимости за аренду
	Процедура РасчитатьОкупаемость()	Во время изменения процедур выше и рассчитывает окупаемость комплектующих
Модуль формы элемента справочника «Договоры»	Процедура ПриОткрытии(Отказ)	Процедура выполняет автоматическое заполнение наименования если оно пустое при открытии
	Процедура ПередЗакрытием(Отказ, ЗавершениеРаботы, ТекстПредупреждения, СтандартнаяОбработка)	Процедура выполняет автоматическое заполнение наименования если оно пустое перед закрытием
	Процедура ПредметДоговораИУслугНаименованиеПриИзменении(Элемент)	При изменении договора или услуги в поле цена падает цена на договор и услугу
	Функция ПолучитьЦену(Наименование)	Функция возвращает цену на договор и услугу
	Процедура ОтображатьЗавершённыеПриИзменении(Элемент)	Процедура переключает видимость завершённых документов
	Процедура ПереключитьОтображениеЗавершённыхДоговоров(ОтображатьЗавершённые)	В процедуре меняется статус
	Процедура ПриОткрытии(Отказ)	При открытии меняется статус

Продолжение таблицы 9

Объект	Функция/Процедура	Описание
Модуль формы документа «Дап. услуги»	Процедура ДоговорПриИзменении(Элемент)	При изменении договора процедура проверяет её дату, если дата оказалась меньше дате создания документа, тогда выдаётся ошибка
	Функция ПроверитьДатуДоговора()	Функция возвращает дату окончания договора
	Процедура ПередЗаписьюНаСервере(Отказ, ТекущийОбъект, ПараметрыЗаписи)	Перед записью договора процедура проверяет её дату, если дата оказалась меньше дате создания документа, тогда выдаётся ошибка
Модуль формы документа «Акт Выполненных работ»	Процедура ВывестиЗаЭтотМесяц(Команда)	Процедура проверяет заполненность года и месяца, если всё заполненно, то процедура заполняет ТЧ
	Процедура ЗаполнитьТЧ()	Процедура напoлняет таблицу значений
	Процедура ПолучитьВсехКлиентовНаОплату()	Процедура выполняет запрос из благодаря которому идёт запись в табличную часть полей клиент, наименование работ и цены всех клиентов, которые должны заплатить
	Процедура ЗаполнитьДерево()	Процедура проводит анализ для каждого клиента на выполнение их работ
	Функция ПолучитьДанныеПоКлиентуИзТЧ(Клиент)	Функция возвращает Клиента из актов
	Функция ПолучитьМесяц()	Функция возвращает номер месяца
	Процедура ПриЗаписиНаСервере(Отказ, ТекущийОбъект, ПараметрыЗаписи)	Процедура выводит ошибку в том случае, если данный документ уже существует
	Функция ПроверитьОплату()	Функция возвращает значение Истина/Ложь при проверке запроса к списку всех документов
Модуль объекта документа «Акт выполненных работ»	Процедура ОбработкаПроведения(Отказ, Режим)	Осуществляет движение расхода по Регистру «ДолгиКлеинтов» путём записи в регистр новых записей
Модуль объекта документа «Оплата»	Процедура ОбработкаПроведения(Отказ, Режим)	Осуществляет движение прихода по Регистру «ДолгиКлеинтов» путём записи в регистр новых записей

Код процедур и функций представлен в приложении А.

4.3 Тестирование информационной системы

Тестирование – это процесс проверки системы на работоспособность и проверки того, что ожидаемый результат оказался реальным, то есть процесс оценки фактического поведения программы.

Основные цели тестирования, следующие:

- проверка соответствия разработанного ПО заявленным функциональным требованиям;
- обнаружение дефектов и устранение потенциальных проблем на начальных стадиях жизненного цикла проекта, предотвращая таким образом увеличение затрат на исправление ошибок позже;
- выявление непредусмотренных сценариев использования системы, позволяющее взглянуть на продукт глазами конечного пользователя.

Таким образом, тестирование обеспечивает уверенность в надежности и качестве программного решения перед его выпуском пользователям.

4.3.1 Функциональное тестирование

Функциональное тестирование осуществляется для оценки соответствия функционала приложения заявленным требованиям и ожиданиям пользователей. Его цель – удостовериться, что приложение функционирует правильно и реализует необходимые функции.

При проведении функционального тестирования отдельно тестируются различные функции программы, тогда как интеграционное тестирование направлено на проверку взаимодействия между компонентами всей системы.

В рамках разработки информационной системы для расчета заработной платы были проведены функциональные тесты (см. таблицу 10). Эти испытания включали проверку процессов создания и обработки документов, формирования отчетов и печатных форм. Итоги функционального тестирования представлены в приложении В.

Таблица 10 – Функциональное тестирование работы системы

№	Название теста	Предполагаемый результат	Полученный результат	Результат тестирования
1	Создание документа «Доп. услуги»	Открыть документ «Доп. услуги», заполнить все реквизиты и табличную часть, нажать на команду «Провести и закрыть. Документ проведётся и выведет список документов	Соответствует ожидаемым результатам	Тест пройден (Рисунок В.1, В.2)
	Создать документ «Акт выполненных работ»	Открыть документ «Доп. услуги», заполнить все реквизиты и табличную часть, нажать на команду «Провести и закрыть. Документ проведётся и выведет список документов	Соответствует ожидаемым результатам	Тест пройден (Рисунок В.3, В.4)
	Создать документ «Оплата»	Открыть документ «Доп. услуги», заполнить все реквизиты и табличную часть, нажать на команду «Провести и закрыть. Документ проведётся и выведет список документов	Соответствует ожидаемым результатам	Тест пройден (Рисунок В.5, В.6)
	Сформировать печатную форму документа «Акт выполненных работ»	Выбрать нужный документ из списка всех документов «Акт выполненных работ», нажать печать. Выведется печатная форма, разбитая на клиентов.	Соответствует ожидаемым результатам	Тест пройден (Рисунок В.7)
	Сформировать отчёт «Задолжности клиентов»	При нажатии кнопки сформировать появится список с клиентами.	Соответствует ожидаемым результатам	Тест пройден (Рисунок В8)

Был проведен комплекс функциональных тестов для модуля расчета оплаты услуг аренды в рамках разработки биллинговой системы проекта «1С: Аренда». Результаты тестирования показывают, что система работает стабильно, отсутствуют очевидные дефекты и проблемы в процессе обработки данных.

4.3.2 Тестирование чёрного ящика

Тестирование – процесс, направленный на оценку и повышение качества программного продукта путем выявления ошибок и недостатков. Обычно оно осуществляется методом обнаружения сбоев и неисправностей в работе системы.

Метод функциональной диаграммы применяется для тестирования запросов студентов следующим образом: пользователи либо выбирают готовый вопрос из предложенных вариантов, либо формулируют собственный, следуя определенным критериям. Таким образом, выбор между вопросами исключается автоматически (например, нельзя одновременно выбрать П1 и П2).

При тестировании по принципу «черного ящика» проверяются различные варианты вводимых данных, чтобы удостовериться в правильной обработке системой. Например, для проверки формата даты типа «ДД.ММ.ГГГГ» используются разные сценарии и методы тестирования.

Таблица 11 – Классы эквивалентности: Табличное представление

	Правильные КЭ	Неправильные КЭ
Диапазон приемлемой даты	01.01.1900 – 31.12.2050 (1)	$-\infty$ -31.12.1899 по 01.01.2051+1 $-\infty$ (2)
Структура даты	ДД.ММ.ГГГГ (3) ДД месяц ГГГГ (4)	ДД миесиц ГГГГ (5) ДД мес ГГГГ (6) ДД.М.ГГГГ (7) ДД.ММ.ГГ (8)
Специальные символы или пробелы	ДД месяц ГГГГ (9)	Наличие лишних пробелов: «ДД месяц ГГГГ» (10) ДД_месяц_ГГГГ (11)

Рассмотрим, тестирование, которое поможет удостовериться, что поле «Дата» корректно принимает даты в нужном формате и отклоняет все ошибочные варианты.

Таблица 12 – Классы эквивалентности: Графическое представление

№	КЭ	Входные данные	Выходные данные
1	Все правильно: 1, 4, 9	1 января 2025	Успешная
2	Всё правильно 1, 3, 9	1.01.2025	Успешно
3	Один не правильны критерий: 2, 3, 9	1.01.1500	Ошибка: Введено число не времени и пространства
4	1, 3, 8	1.01.25	Ошибка: Введена дата с неправильным и годом
5	Всё неправильно: 2, 8, 11	01:01:15000	Ошибка: Это не дата

Ограничение значений поля типа «Дата» позволяет обеспечить корректность, приавльность и надежность ввода и хранения данных в необходимом типе данных.

Рассмотрим ключевые способы ограничения полей дат и случаи их практического применения. Типы границ полей представлены в таблице 13.

Таблица 13 – Типы границ

Границы	
Физические	Невозможные даты
Логические	Несоответствие календарю
Технологические	Предельные значения типов данных
Произвольные	Требования технического задания

Были проведены всеобъемлющие и необходимы тесты с полем тип которого является «Дата». Результат тестирования с выходными данными преведены в таблице 14.

Таблица 14 – Тесты граничных условий

№	Границы	Входные данные	Выходные данные
1	Физические границы t1	ДД.ММ.ГГГГ	Дата с отрицательным днём Дата отклонена
2	Физические границы	ДД.ММ.	Дата с пустым годом Дата отклонена
3	Физические границы	ДД.00.ГГГГ	Дата с нулевым месяцем Дата отклонена
4	Логические границы	30.02.ГГГГ	Превышающий количество дней в месяце (февраль имеет 28 дней. Високосный год 29 дней) Дата отклонена
5	Технические границы	ДД.ММ.2051	Максимально поддерживаемый год Дата отклонена

Для удобства анализа причин и следствий создадим таблицу решений, где будем учитывать различные комбинации условий создания документа. Данная таблица представлена под номером 15.

Таблица 15 – Таблица решений причин и следствий

Причины	
П1	Номер документа уникален
П2	Дата составления документа – текущая дата
П3	Наименование аренды
П4	Сумма арендной платы
П5	Срок аренды
Следствия	
С1	Ошибка при проверке уникального номера документа
С2	Ошибка при проверке даты составления
С3	Ошибка при проверке наименования арендованного имущества
С4	Ошибка при проверке суммы арендной платы
С5	Ошибка при проверке срока аренды
С6	Реквизиты документа введены корректно Документ проводится

Для упрощения использования алгоритма было проведено комбинирование и попарное тестирование. Результаты тестирования изображены на рисунке 28 и 29.

	× Номер документа	× Дата составления гс	× Наименование аренды	× Сумма аренды	× Срок аренды	+
× Row 1	>0	Текущая дата	[А-Я, а-я, 0-9, A-Z, a-z, ,-,	Отрицательное числ	Текущая дата+	+
× Row 2	<0	Не текущая дата	[А-Я, а-я, 0-9, A-Z, a-z]	Положительное числс	<Текущая дата	+
× Row 3	{your value}	{your value}	[А-Я, а-я, 0-9, ,-,	{your value}	{your value}	+
× Row 4	{your value}	{your value}	[0-9, A-Z, a-z, ,-,	{your value}	{your value}	+
× Row 5	{your value}	{your value}	[А-Я, а-я, A-Z, a-z, ,-,	{your value}	{your value}	+
+	+	+	+	+	+	+

Рисунок 28 – Комбинированное тестирование

	A	B	C	D	E	F
1		Номер документа	Дата составления документа	Наименование аренды	Сумма аренды	Срок аренды
2	1	>0	Текущая дата	[А-Я, а-я, 0-9, A-Z, a-z, ,-,	Отрицательное число	Текущая дата+
3	2	>0	Текущая дата	[А-Я, а-я, 0-9, A-Z, a-z, ,-,	Отрицательное число	<Текущая дата
4	3	>0	Текущая дата	[А-Я, а-я, 0-9, A-Z, a-z, ,-,	Положительное число	Текущая дата+
5	4	>0	Текущая дата	[А-Я, а-я, 0-9, A-Z, a-z, ,-,	Положительное число	<Текущая дата
6	5	>0	Текущая дата	[А-Я, а-я, 0-9, A-Z, a-z]	Отрицательное число	Текущая дата+
7	6	>0	Текущая дата	[А-Я, а-я, 0-9, A-Z, a-z]	Отрицательное число	<Текущая дата
8	7	>0	Текущая дата	[А-Я, а-я, 0-9, A-Z, a-z]	Положительное число	Текущая дата+
9	8	>0	Текущая дата	[А-Я, а-я, 0-9, A-Z, a-z]	Положительное число	<Текущая дата
10	9	>0	Текущая дата	[А-Я, а-я, 0-9, ,-,	Отрицательное число	Текущая дата+
11	10	>0	Текущая дата	[А-Я, а-я, 0-9, ,-,	Отрицательное число	<Текущая дата
12	11	>0	Текущая дата	[А-Я, а-я, 0-9, ,-,	Положительное число	Текущая дата+
13	12	>0	Текущая дата	[А-Я, а-я, 0-9, ,-,	Положительное число	<Текущая дата
14	13	>0	Текущая дата	[0-9, A-Z, a-z, ,-,	Отрицательное число	Текущая дата+
15	14	>0	Текущая дата	[0-9, A-Z, a-z, ,-,	Отрицательное число	<Текущая дата
16	15	>0	Текущая дата	[0-9, A-Z, a-z, ,-,	Положительное число	Текущая дата+
17	16	>0	Текущая дата	[0-9, A-Z, a-z, ,-,	Положительное число	<Текущая дата
18	17	>0	Текущая дата	[А-Я, а-я, A-Z, a-z, ,-,	Отрицательное число	Текущая дата+
19	18	>0	Текущая дата	[А-Я, а-я, A-Z, a-z, ,-,	Отрицательное число	<Текущая дата
20	19	>0	Текущая дата	[А-Я, а-я, A-Z, a-z, ,-,	Положительное число	Текущая дата+
21	20	>0	Текущая дата	[А-Я, а-я, A-Z, a-z, ,-,	Положительное число	<Текущая дата
22	21	>0	Не текущая дата	[А-Я, а-я, 0-9, A-Z, a-z, ,-,	Отрицательное число	Текущая дата+
23	22	>0	Не текущая дата	[А-Я, а-я, 0-9, A-Z, a-z, ,-,	Отрицательное число	<Текущая дата
24	23	>0	Не текущая дата	[А-Я, а-я, 0-9, A-Z, a-z, ,-,	Положительное число	Текущая дата+
25	24	>0	Не текущая дата	[А-Я, а-я, 0-9, A-Z, a-z, ,-,	Положительное число	<Текущая дата
26	25	>0	Не текущая дата	[А-Я, а-я, 0-9, A-Z, a-z]	Отрицательное число	Текущая дата+
27	26	>0	Не текущая дата	[А-Я, а-я, 0-9, A-Z, a-z]	Отрицательное число	<Текущая дата
28	27	>0	Не текущая дата	[А-Я, а-я, 0-9, A-Z, a-z]	Положительное число	Текущая дата+
29	28	>0	Не текущая дата	[А-Я, а-я, 0-9, A-Z, a-z]	Положительное число	<Текущая дата
30	29	>0	Не текущая дата	[А-Я, а-я, 0-9, ,-,	Отрицательное число	Текущая дата+
31	30	>0	Не текущая дата	[А-Я, а-я, 0-9, ,-,	Отрицательное число	<Текущая дата
32	31	>0	Не текущая дата	[А-Я, а-я, 0-9, ,-,	Положительное число	Текущая дата+
33	32	>0	Не текущая дата	[А-Я, а-я, 0-9, ,-,	Положительное число	<Текущая дата
34	33	>0	Не текущая дата	[0-9, A-Z, a-z, ,-,	Отрицательное число	Текущая дата+
35	34	>0	Не текущая дата	[0-9, A-Z, a-z, ,-,	Отрицательное число	<Текущая дата
36	35	>0	Не текущая дата	[0-9, A-Z, a-z, ,-,	Положительное число	Текущая дата+
37	36	>0	Не текущая дата	[0-9, A-Z, a-z, ,-,	Положительное число	<Текущая дата
38	37	>0	Не текущая дата	[А-Я, а-я, A-Z, a-z, ,-,	Отрицательное число	Текущая дата+
39	38	>0	Не текущая дата	[А-Я, а-я, A-Z, a-z, ,-,	Отрицательное число	<Текущая дата
40	39	>0	Не текущая дата	[А-Я, а-я, A-Z, a-z, ,-,	Положительное число	Текущая дата+
41	40	>0	Не текущая дата	[А-Я, а-я, A-Z, a-z, ,-,	Положительное число	<Текущая дата

Рисунок 29 – Результаты тестирования

Был собран набор тестов, результат представлен в таблице 16.

Таблица 16 – Набор тестов

№	Входные данные	Выходные данные	Фактический результат
1	Номер документа уникален:0000 Дата составления документа:20.03.2027 Наименование аренды: Введите наименование Сумма арендной платы: -10 000 Срок аренды: .	Не верно указан номер документа. Дата не может быть будущей. Наименование не указано. Сумма арендной платы не может иметь отрицательное значение. Срок арендной платы не может быть пустым.	
2	Номер документа уникален: 0001 Дата составления документа: 20.03.2027 Наименование аренды: Введите наименование Сумма арендной платы: -10 000 Срок аренды: .	Дата не может быть будущей. Наименование не указано. Сумма арендной платы не может иметь отрицательное значение. Срок арендной платы не может быть пустым.	
3	Номер документа уникален:0000 Дата составления документа:20.03.2025 Наименование аренды: Введите наименование Сумма арендной платы: -10 000 Срок аренды: .	Не верно указан номер документа. Наименование не указано. Сумма арендной платы не может иметь отрицательное значение. Срок арендной платы не может быть пустым.	
4	Номер документа уникален:0000 Дата составления документа:20.03.2027 Наименование аренды: «1С:Предприятие 8.3» Сумма арендной платы: -10 000 Срок аренды: .	Не верно указан номер документа. Дата не может быть будущей. Сумма арендной платы не может иметь отрицательное значение. Срок арендной платы не может быть пустым.	
5	Номер документа уникален:0000 Дата составления документа:20.03.2027 Наименование аренды: Введите наименование Сумма арендной платы: 10 000 Срок аренды: .	Не верно указан номер документа. Дата не может быть будущей. Наименование не указано. Срок арендной платы не может быть пустым.	
6	Номер документа уникален:0000 Дата составления документа:20.03.2027 Наименование аренды: Введите наименование Сумма арендной платы: -10 000 Срок аренды: 23.03.2025 – 23.03.2026	Не верно указан номер документа. Дата не может быть будущей. Наименование не указано. Сумма арендной платы не может иметь отрицательное значение.	

Продолжение таблицы 16

№	Входные данные	Выходные данные	Фактический результат
7	Номер документа уникален:0001 Дата составления документа:20.03.2025 Наименование аренды: Введите наименование Сумма арендной платы: -10 000 Срок аренды: .	Наименование не указано. Сумма арендной платы не может иметь отрицательное значение. Срок арендной платы не может быть пустым.	
8	Номер документа уникален:0001 Дата составления документа:20.03.2025 Наименование аренды: «1С:Предприятие 8.3» Сумма арендной платы: -10 000 Срок аренды: .	Сумма арендной платы не может иметь отрицательное значение. Срок арендной платы не может быть пустым.	
9	Номер документа уникален:0001 Дата составления документа:20.03.2025 Наименование аренды: «1С:Предприятие 8.3» Сумма арендной платы: 10 000 Срок аренды: .	Срок арендной платы не может быть пустым.	
10	Номер документа уникален:0001 Дата составления документа:20.03.2025 Наименование аренды: Введите наименование Сумма арендной платы: 10 000 Срок аренды: 23.03.2025 – 23.03.2026	Наименование не указано.	
11	Номер документа уникален:0001 Дата составления документа:20.03.2025 Наименование аренды: «1С:Предприятие 8.3» Сумма арендной платы: -10 000 Срок аренды: 23.03.2025 – 23.03.2026	Сумма арендной платы не может иметь отрицательное значение.	
12	Номер документа уникален:0001 Дата составления документа:20.03.2027 Наименование аренды: «1С:Предприятие 8.3» Сумма арендной платы: 10 000 Срок аренды: 23.03.2025 – 23.03.2026	Дата не может быть будущей.	
13	Номер документа уникален:0000 Дата составления документа:20.03.2025 Наименование аренды: «1С:Предприятие 8.3» Сумма арендной платы: 10 000 Срок аренды: 23.03.2025 – 23.03.2026	Не верно указан номер документа	

Окончание таблицы 16

№	Входные данные	Выходные данные	Фактический результат
14	Номер документа уникален:0001 Дата составления документа:20.03.2025 Наименование аренды: «1С:Предприятие 8.3» Сумма арендной платы: 10 000 Срок аренды: 23.03.2025 – 23.03.2026	Документ успешно сгенерирован	

Выполнено тестирование методом «чёрного ящика», которое подтвердило полную готовность системы к эксплуатации.

4.3.2 Метрика Холстеда

Метрики программного обеспечения, предложенные Морисом Холстедом, представляют собой набор характеристик, используемых для измерения сложности программ. Основываясь на количестве уникальных операторов и операндов, эти метрики оценивают затраты ресурсов на разработку и сопровождение программы. Основные показатели включают длину программы, её лексический состав, общий объем, уровень сложности и требуемые усилия.

Анализ этих данных позволяет программистам объективно оценивать сложность продукта, находить участки для улучшения и улучшать качество ПО. Согласно теории Холстеда, любая компьютерная программа состоит из набора элементов (токенов), подразделяемых на две категории: операторы и операнды. Операторы определяют выполняемые операции, тогда как операнды выступают объектами действий. Оценивая количество и разнообразие этих токенов, метрики Холстеда предоставляют возможность измерять степень сложности, трудоёмкости и общего уровня качества исходного кода.

В «Метриках программного обеспечения» Холстеда:

- n_1 – количество различных операторов;
- n_2 – количество различных операндов;
- N_1 – общее количество вхождений операторов;

- N_2 – общее количество вхождений операндов.

Используя данные характеристики, можно рассчитать следующие показатели:

- длина фрагмента программы, которая рассчитывается по формуле

$$N = N_1 + N_2;$$

- словарь фрагмента программы, который рассчитывается по формуле

$$n = n_1 + n_2;$$

- объем фрагмента программы, который рассчитывается по формуле

$$V = N \times \log_2 n;$$

- сложность фрагмента программы, которая рассчитывается по формуле

$$HDiff = (n_1 \div 2) \times (N_2 \div n_2).$$

Для процедуры «ЗаполнитьДерево()» из модуля формы документа «Акт выполненных работ» был произведён расчёт количественных показателей с использованием метрики Холстеда. Код процедуры приведён в листинге 1.

Листинг 1 – Код процедуры «ЗаполнитьДерево»

```
Процедура ЗаполнитьДерево()
МассивКлиентов = Новый Массив;
Для Каждого Строка Из Объект.Акты Цикл
    Если МассивКлиентов.Найти(Строка.Клиент) = Неопределено Тогда
        МассивКлиентов.Добавить(Строка.Клиент);
    КонецЕсли;
КонецЦикла;
НовоеДерево = РеквизитФормыВЗначение("Дерево");
Для Каждого СтрокаКорня Из МассивКлиентов Цикл
    ЦенаДляКорня = 0;
    Корень = НовоеДерево.Строки.Добавить();
    Корень.ДеревоКлиент = СтрокаКорня;
    Для Каждого Строка Из
        ПолучитьДанныеПоКлиентуИзТЧ(СтрокаКорня) Цикл
        Ветка = Корень.Строки.Добавить();
        Ветка.ДеревоНаименованиеРабот = Строка.НаименованиеРабот;
        Ветка.ДеревоЦена = Строка.Цена;
        ЦенаДляКорня = ЦенаДляКорня + Строка.Цена;
    КонецЦикла;
    Корень.ДеревоЦена = ЦенаДляКорня;
```

КонецЦикла;

ЗначениеВРеквизитФормы(НовоеДерево, "Дерево");

КонецПроцедуры

Все операторы, используемые в процедуре «ЗаполнитьДерево» представленный в таблице 17.

Таблица 17 – Операторы процедуры

Название	Количество
=	4
.	13
Если	1
КонецЕсли	1
ДляКаждого	3
Цикл	3
КонецЦикла	3
Новый	1
Найти	1
Добавить	1
Присвоение цены	1
ЗначениеВРеквизитФормы	1
РеквизитФормыВЗначение	1
Строки.Добавить	2
+	1

Все операнды, используемые в процедуре «ЗаполнитьДерево», представленный в таблице 18.

Таблица 18 – Операнды процедуры

Название	Количество
МассивКлиентов	3
Объект.Акты	1
Строка	3
Строка.Клиент	2
НовоеДерево	2
СтрокаКорня	2
ЦенаДляКорня	2
Корень	3
Ветка	2
Строка.НаименованиеРабот	1
Строка.Цена	2
Неопределено	1
Дерево	1
Получение данных клиента	1
Ручное преобразование реkv.	1

После подсчёта количества операторов и операндов были рассчитаны значения перечисленных выше показателей:

- длина программы (N): 62;
- словарь программы (n): 30;
- объём программы (V): 304.23;
- сложность программы (HDiff): 13.

Таким образом был выполнен расчёт количественных показателей процедуры «ЗаполнитьДерево» с использованием метрики Холстеда.

Заключение

Данная выпускная квалификационная работа подтвердила важность создания системы для работы компании по оказанию услуг аренды серверного оборудования. После того как были изучены аналоги, и на основе сравнительных критериев были выделены основные требования к создаваемой системе, были выполнены мероприятия по проектированию и разработке новой системы.

При проектировании новой системы был проведён анализ бизнес-процессов и построены различные диаграммы, которые отображают тот или иной аспект системы. Была построена функциональная модель IDEF0, спроектированы UML диаграммы и произведено описание этой диаграммы. При выполнении данной выпускной квалификационной работы так же было проведено планирование и расчёт бюджета новой разрабатываемой системы.

При разработке была использовалась информационная платформа 1С: Предприятие 8.3, которая позволила разработать системы для помощи компании по оказанию услуг аренды серверного оборудования. Данная системы облегчается и автоматизирует работу такой компании, позволяя сотрудникам сосредоточиться не на поддержании работоспособности системы, а на продвижении всей компании в целом. Так же данная системы позволяет проводить анализ данных, и данную систему можно спокойно масштабировать благодаря гибкости выбранной платформы 1С: Предприятие 8.3.

Список использованных источников

1. Петров А.В., Иванова Е.С. Основы проектирования информационных систем. Москва: Просвет, 2022.
2. «1С: Аренда и управление персоналом 8»: комплексное решение для бизнеса. [Электронный ресурс]URL: <https://v8.1c.ru/hrm/>? (Дата обращения 23.05.2025).
3. Матвеев Д.Г. Методология объектно-ориентированного анализа и проектирования. СПб.: Наука и образование, 2021.
4. Полезные практики CI/CD в DevOps среде. [Электронный ресурс]URL: <https://habr.com/ru/articles/549054> (Дата обращения 23.05.2025).
5. Ильин М.Н. Анализ данных и создание отчетов. Екатеринбург: Университетская книга, 2020.
6. Советы по повышению эффективности командной работы над проектами. [Электронный ресурс]URL: <https://www.agilerussia.ru/blog/knowledge-base/project-team-work-effectiveness> (Дата обращения 23.05.2025).
7. Зими́на Т.А. Управление рисками в ИТ-проектировании. Новосибирск: Академия печати, 2019.
8. Модели проектирования информационных систем. [Электронный ресурс]URL: <https://trinion.org/blog/information-system-design-models> (Дата обращения 23.05.2025).
9. Коваленко В.П. Современные методы автоматизированного тестирования. Самара: Самарский университет, 2018.
10. Улучшаем качество программного обеспечения с использованием метрик. [Электронный ресурс]URL: <https://geeksforgeeks.org/software-engineering-halsteads-software-metrics> (Дата обращения 23.05.2025).
11. Гришина Л.И. Организация эффективного документооборота. Челябинск: Южно-Уральский институт менеджмента, 2017.

12. Интеграция и настройка инструментов DevOps. [Электронный ресурс]URL: <https://loadimpact.com/blog/integrating-devops-tools> (Дата обращения 23.05.2025).
13. Сидоренко Н.К. Практика внедрения принципов Scrum. Владивосток: Дальневосточный университет, 2016.
14. Лучшие методики оптимизации баз данных. [Электронный ресурс]URL: <https://sqlshack.com/database-optimization-best-practices> (Дата обращения 23.05.2025).
15. Романов Ю.Ф. Моделирование бизнес-процессов организации. Ярославль: ЯрГУ, 2015.
16. Основы проектирования UI/UX интерфейсов. [Электронный ресурс]URL: <https://medium.com/@alexandernikulin/ui-ux-design-basics-c216d9155bdd> (Дата обращения 23.05.2025).
17. Волкова Р.Е. Принципы качественного управления требованиями. Иркутск: Байкальский госуниверситет, 2014.
18. Изучение подходов к архитектуре больших данных. [Электронный ресурс]URL: <https://bigdataexpert.ru/approaches-to-big-data-architecture> (Дата обращения 23.05.2025).
19. Барановский П.Л. Автоматизация процессов планирования ресурсов. Саратов: Саратовский государственный технический университет, 2013.
20. Методы контроля качества программного обеспечения. [Электронный ресурс]URL: <https://www.quality-assurance-expert.com/qa-methodologies-overview> (Дата обращения 23.05.2025).
21. Воронов В.Б. Разработка клиент-серверных приложений. Ростов-на-Дону: Южный федеральный университет, 2012.
22. Подходы к созданию высокоэффективных API-интерфейсов. [Электронный ресурс]URL: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Web_Frameworks/API_Design_Patterns (Дата обращения 23.05.2025).

23. Тихонов А.Ю. Оптимизация загрузки страниц в web-разработке. Хабаровск: Тихоокеанский госуниверситет, 2011.
24. Инструменты мониторинга инфраструктуры и безопасности. [Электронный ресурс]URL: <https://blog.gitguardian.com/server-security-monitoring-tools> (Дата обращения 23.05.2025).
25. Макаров С.Д. Создание эффективных моделей взаимодействия пользователя и системы. Нижний Новгород: Нижегородский госуниверситет, 2010.
26. Базовые концепции DevSecOps и безопасность в процессе разработки. [Электронный ресурс]URL: <https://cloudsecurityalliance.org/assets/docs/research/DevSecOps-Basics-and-Safeguards.pdf> (Дата обращения 23.05.2025).
27. Осипов А.Н. Применение контейнеризации в распределённых системах. Омск: Омский политехнический университет, 2009.
28. Определение и классификация технического долга. [Электронный ресурс]URL: <https://devblogs.microsoft.com/devops/technical-debt-minimization-strategies> (Дата обращения 23.05.2025).
29. Павловский Е.В. Практики работы с большими объёмами данных. Воронеж: Воронежский госуниверситет, 2008.
30. Международные стандарты качества программного обеспечения (ISO/IEC 25010). [Электронный ресурс]URL: <https://www.iso.org/standard/61433.html> (Дата обращения 23.05.2025).

Приложение А
(обязательное)
Листинг программы

Листинг А.1 – Код модуля формы справочника «Договоры»

```

&НаКлиенте
Процедура ОтображатьЗавершенныеПриИзменении (Элемент)

    ПереключитьОтображениеЗавершенныхДоговоров (ОтображатьЗавершенн
ые) ;

КонецПроцедуры

&НаКлиенте
Процедура
ПереключитьОтображениеЗавершенныхДоговоров (ОтображатьЗавершенные)

    ОбщегоНазначенияКлиентСервер.УстановитьЭлементОтбораДинамическ
огоСписка (
        Список, "Статус",
        ПредопределенноеЗначение ("Перечисление.СтатусДоговора.Действует") ,
        , НЕ ОтображатьЗавершенные) ;

КонецПроцедуры

&НаКлиенте
Процедура ПриОткрытии (Отказ)

    ПереключитьОтображениеЗавершенныхДоговоров (ОтображатьЗавершенн
ые) ;

КонецПроцедуры

```

Листинг А.2 – Код модуля формы справочника «Сервера»

```

&НаКлиенте
Процедура ПередЗаписью (Отказ, ПараметрыЗаписи)

    Если ПроверитьНаименованиеНаДубликат () = Ложь Тогда
        Отказ = Истина;
        Предупреждение ("Сервер с таким именем уже существует!");
    КонецЕсли;

КонецПроцедуры

&НаСервере
Функция ПроверитьНаименованиеНаДубликат ()

    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |     Сервера.Ссылка КАК Ссылка
        | ИЗ
        |     Справочник.Сервера КАК Сервера
        | ГДЕ
        |     Сервера.Наименование = &Наименование
        |     И Сервера.Ссылка <> &Ссылка";

```

```

        Запрос.УстановитьПараметр ("Наименование",
Объект.Наименование) ;
        Запрос.УстановитьПараметр ("Ссылка", Объект.Ссылка) ;

        РезультатЗапроса = Запрос.Выполнить () ;

        Если РезультатЗапроса.Пустой () Тогда
            Возврат Истина
        Иначе
            Возврат Ложь
        КонецЕсли;

```

КонецФункции

```

&НаСервере
Функция ПолучитьЦенуВМесяц (Наименование)

```

```

    Возврат
    Справочники.Комплектующие.НайтиПоНаименованию (Наименование.Наимено-
вание) .ПредварительнаяСтоимостьВАренде;

```

КонецФункции

Листинг А.3 – Код модуля формы справочника «Комплектующие»

```

&НаКлиенте
Процедура СтоимостьЗакупкиПриИзменении (Элемент)

```

```

    РассчитатьОкупаемость () ;

```

КонецПроцедуры

```

&НаКлиенте
Процедура ПредварительнаяСтоимостьВАрендеПриИзменении (Элемент)

```

```

    РассчитатьОкупаемость () ;

```

КонецПроцедуры

```

&НаКлиенте
Процедура РассчитатьОкупаемость ()

```

```

    Если Объект.СтоимостьЗакупки <> 0 и
    Объект.ПредварительнаяСтоимостьВАренде <> 0 Тогда
        Объект.Окупаемость = Объект.СтоимостьЗакупки /
    Объект.ПредварительнаяСтоимостьВАренде;
    КонецЕсли;

```

КонецПроцедуры

Листинг А.4 – Код модуля формы справочника «Договоры»

&НаКлиенте
Процедура ПриОткрытии(Отказ)

```
Если Объект.Наименование = "" Тогда
    Объект.Наименование = Объект.Код;
КонецЕсли;
```

КонецПроцедуры

&НаКлиенте
Процедура ПередЗакрытием(Отказ, ЗавершениеРаботы,
ТекстПредупреждения, СтандартнаяОбработка)

```
Если Объект.Наименование = "" Тогда
    Объект.Наименование = Объект.Код;
КонецЕсли;
```

КонецПроцедуры

&НаКлиенте
Процедура ПредметДоговораИУслугНаименованиеПриИзменении(Элемент)

```
СтрокаТЧ = Элементы.ПредметДоговораИУслуг.ТекущиеДанные;
СтрокаТЧ.Цена = ПолучитьЦену(СтрокаТЧ.Наименование);
```

КонецПроцедуры

&НаСервере
Функция ПолучитьЦену(Наименование)

```
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ
                |
                СУММА (СервераТехническиеХарактеристики.ЦенаВМесяц) КАК
ЦенаВМесяц
                | ИЗ
                |
                Справочник.Сервера.ТехническиеХарактеристики КАК
СервераТехническиеХарактеристики
                | ГДЕ
                |     СервераТехническиеХарактеристики.Ссылка =
&Ссылка";
```

```
Запрос.УстановитьПараметр ("Ссылка", Наименование);
```

```
Возврат Запрос.Выполнить().Выгрузить().Итог("ЦенаВМесяц");
```

КонецФункции

Листинг А.5 – Код модуля формы документа «Доп услуги»

&НаКлиенте
Процедура ДоговорПриИзменении(Элемент)

```

        Если ПроверитьДатуДоговора() < Объект.Дата Тогда
            Сообщить ("Дата окончания договора меньше даты документа,
проведение будет невозможно!");

        КонецЕсли;

КонецПроцедуры

&НаСервере
Функция ПроверитьДатуДоговора()

    Возврат Объект.Договор.КонечнаяДатаДоговора;

КонецФункции

&НаСервере
Процедура ПередЗаписьюНаСервере(Отказ, ТекущийОбъект,
ПараметрыЗаписи)

    Если ПроверитьДатуДоговора() < Объект.Дата Тогда

        Отказ = Истина;
        Сообщить ("Дата окончания договора меньше даты документа,
проведение будет невозможно!");

    КонецЕсли;

КонецПроцедуры

```

Листинг А.6 – Код модуля формы документа «Акт выполненных работ»

```

&НаКлиенте
Процедура ВывестиЗаЭтотМесяц(Команда)

    Если Объект.Год = 0 Или Объект.Месяц = "" Тогда
        Предупреждение("Сначала выберите год и месяц!");
    Иначе
        ЗаполнитьТЧ();
    КонецЕсли;

КонецПроцедуры

&НаСервере
Процедура ЗаполнитьТЧ()

    Объект.Акты.Очистить();
    Дерево.ПолучитьЭлементы().Очистить();
    ПолучитьВсехКлиентовНаОплату();
    ЗаполнитьДерево();

КонецПроцедуры

```

&НаСервере

Процедура ПолучитьВсехКлиентовНаОплату()

Дат = СтрЗаменить(Строка(Объект.Год), Символы.НПП, "") +
ПолучитьМесяц() + "01";

```

Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ
|
| ДоговорыПредметДоговораИУслуг.Ссылка.Клиент КАК Клиент,
|      ДоговорыПредметДоговораИУслуг.Наименование
КАК Наименование,
|      СУММА(ДоговорыПредметДоговораИУслуг.Цена)
/ (РАЗНОСТЬДАТ(&НачальнаяДатаДоговора, &КонечнаяДатаДоговора,
ДЕНЬ) + 1) *
(РАЗНОСТЬДАТ(ДоговорыПредметДоговораИУслуг.Ссылка.НачальнаяДатаДог
овора, &КонечнаяДатаДоговора, ДЕНЬ) + 1) КАК Цена
| ИЗ
|      Справочник.Договоры.ПредметДоговораИУслуг
КАК ДоговорыПредметДоговораИУслуг
| ГДЕ
|
| ДоговорыПредметДоговораИУслуг.Ссылка.НачальнаяДатаДоговора
МЕЖДУ &НачальнаяДатаДоговора И &КонечнаяДатаДоговора
|
| СГРУППИРОВАТЬ ПО
|
| ДоговорыПредметДоговораИУслуг.Ссылка.Клиент,
|
| ДоговорыПредметДоговораИУслуг.Наименование,
|
| ДоговорыПредметДоговораИУслуг.Ссылка.НачальнаяДатаДоговора
|
| ОБЪЕДИНИТЬ ВСЕ
|
| ВЫБРАТЬ
|      ДопУслугиУслуги.Ссылка.Договор.Клиент,
|      ДопУслугиУслуги.Наименование,
|      ДопУслугиУслуги.Цена
| ИЗ
|      Документ.ДопУслуги.Услуги КАК
ДопУслугиУслуги
| ГДЕ
|      ДопУслугиУслуги.Ссылка.Дата МЕЖДУ
&НачальнаяДатаДоговора И &КонечнаяДатаДоговора";

```

Запрос.УстановитьПараметр("НачальнаяДатаДоговора", НачалоМесяца
(Дат));

Запрос.УстановитьПараметр("КонечнаяДатаДоговора",
КонецМесяца(Дат));

```

Выборка = Запрос.Выполнить().Выбрать();

Пока Выборка.Следующий() Цикл

    НоваяСтрока = Объект.Акты.Добавить();
    НоваяСтрока.Клиент = Выборка.Клиент;
    НоваяСтрока.НаименованиеРабот =
Выборка.Наименование;
    НоваяСтрока.Цена = Выборка.Цена;

КонецЦикла;

КонецПроцедуры

```

Листинг Б.7 – Код модуля объекта документа «Оплата»

```

Процедура ОбработкаПроведения(Отказ, Режим)
    //{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
    // Данный фрагмент построен конструктором.
    // При повторном использовании конструктора, внесенные вручную
    изменения будут утеряны!!!

    // регистр ДолгиКлиентов Приход
    Движения.ДолгиКлиентов.Записывать = Истина;
    Движение = Движения.ДолгиКлиентов.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Клиент = Клиент;
    Движение.Сумма = Сумма;

    //}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры

```

Листинг Б8 – Код модуля объекта документа «Акт выполненных работ»
 Процедура ОбработкаПроведения(Отказ, Режим)

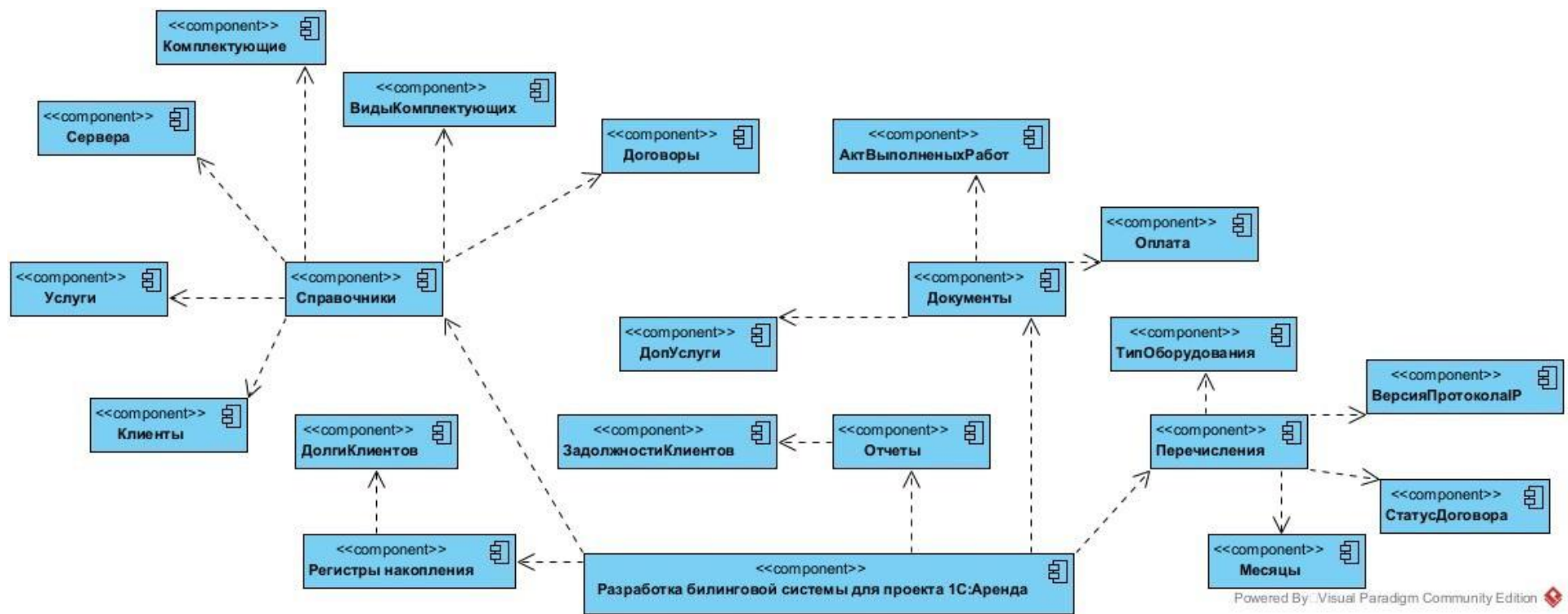
```

    // регистр ДолгиКлиентов Расход
    Движения.ДолгиКлиентов.Записывать = Истина;
    Для Каждого ТекСтрокаАкты Из Акты Цикл
        Движение = Движения.ДолгиКлиентов.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Клиент = ТекСтрокаАкты.Клиент;
        Движение.Сумма = ТекСтрокаАкты.Цена;
    КонецЦикла;

КонецПроцедуры

```

Приложение Б
(рекомендуемое)
Диаграмма компонентов



Приложение В
(рекомендуемое)
Результаты тестирования

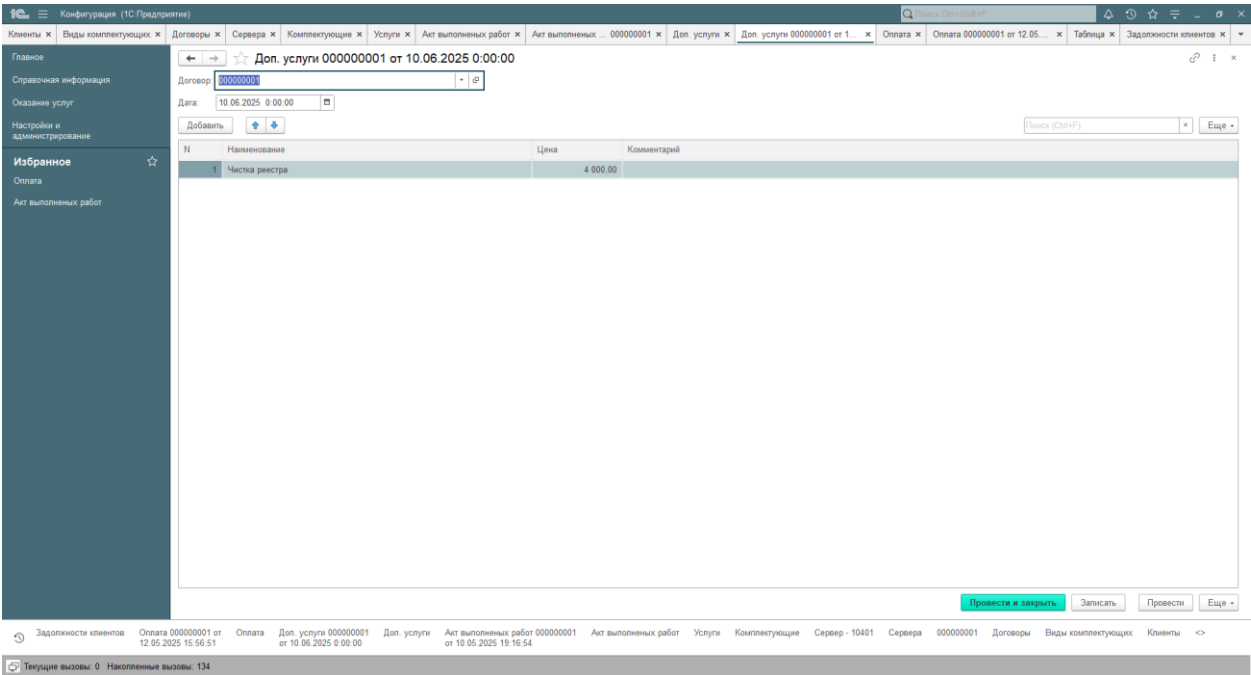


Рисунок В.1 – Создание документа «Доп услуги»

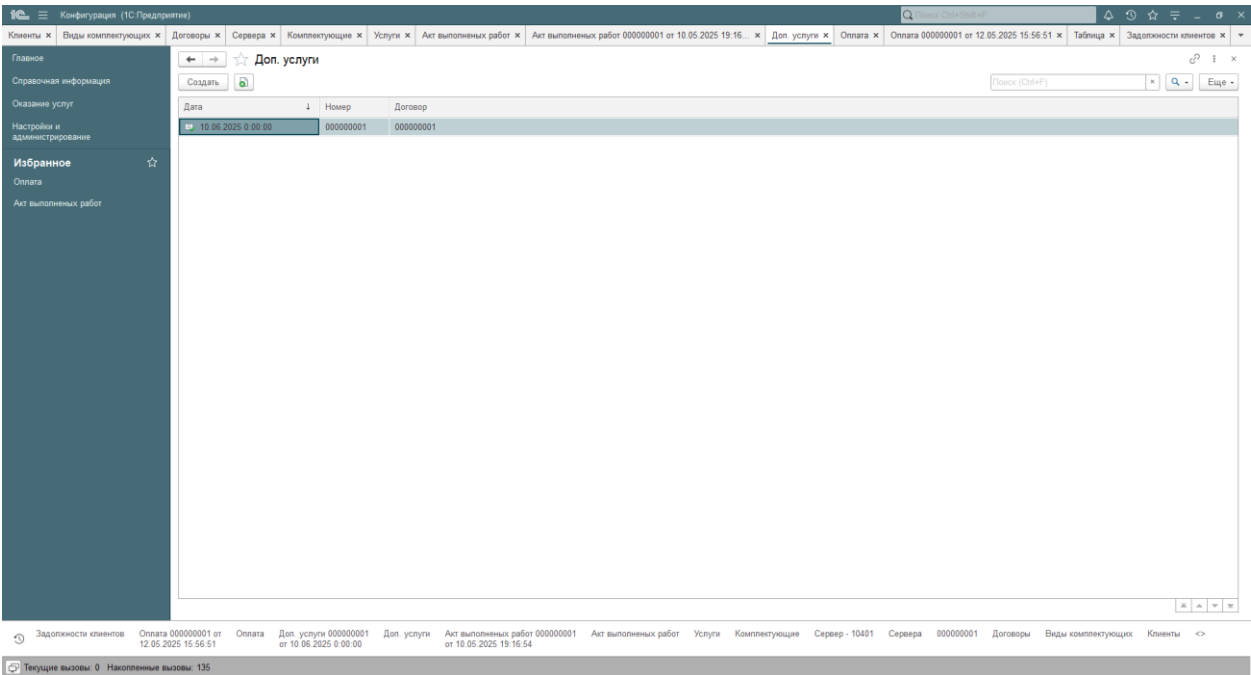


Рисунок В.2 – Форма списка документов «Доп услуги»

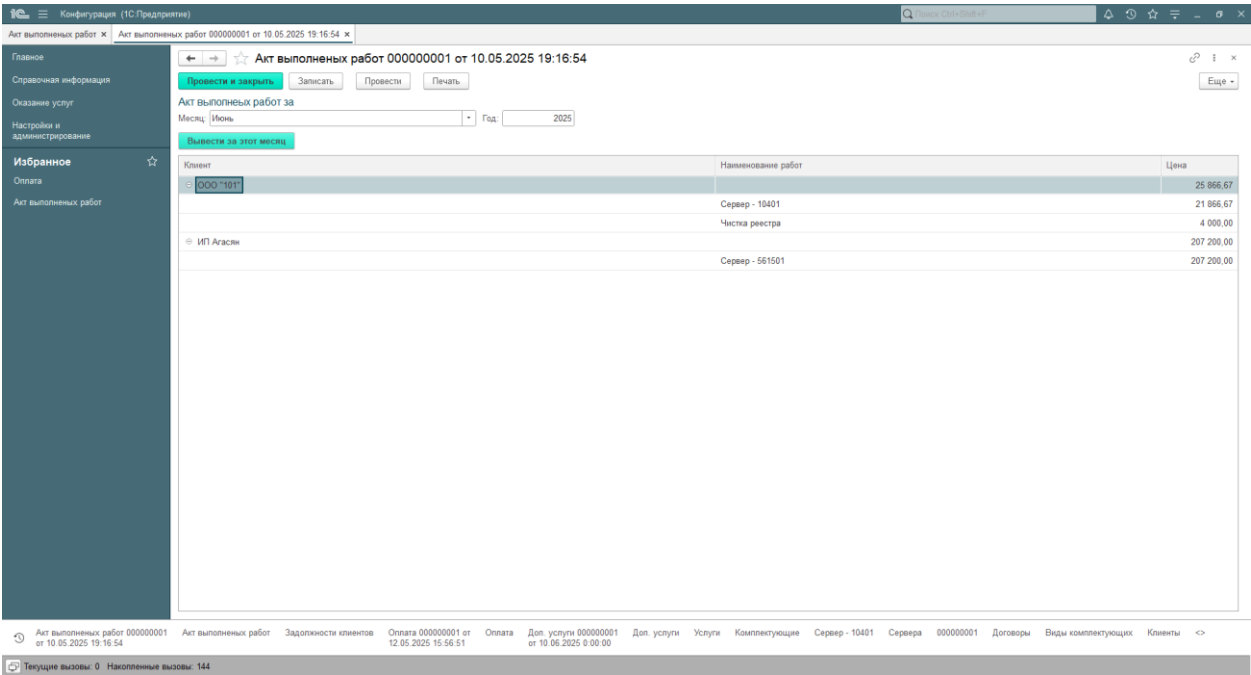


Рисунок В.3 – Создание документа «Акт выполненных работ»

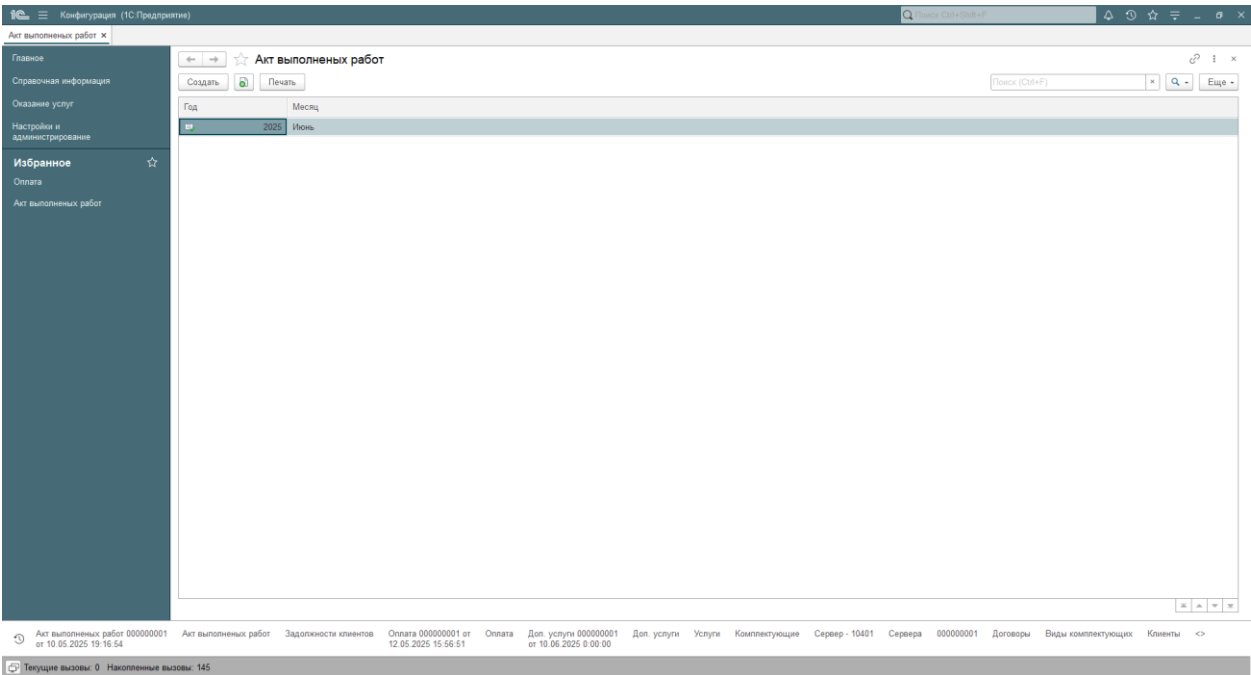


Рисунок В.4 – Форма списка документов «Акт выполненных работ»

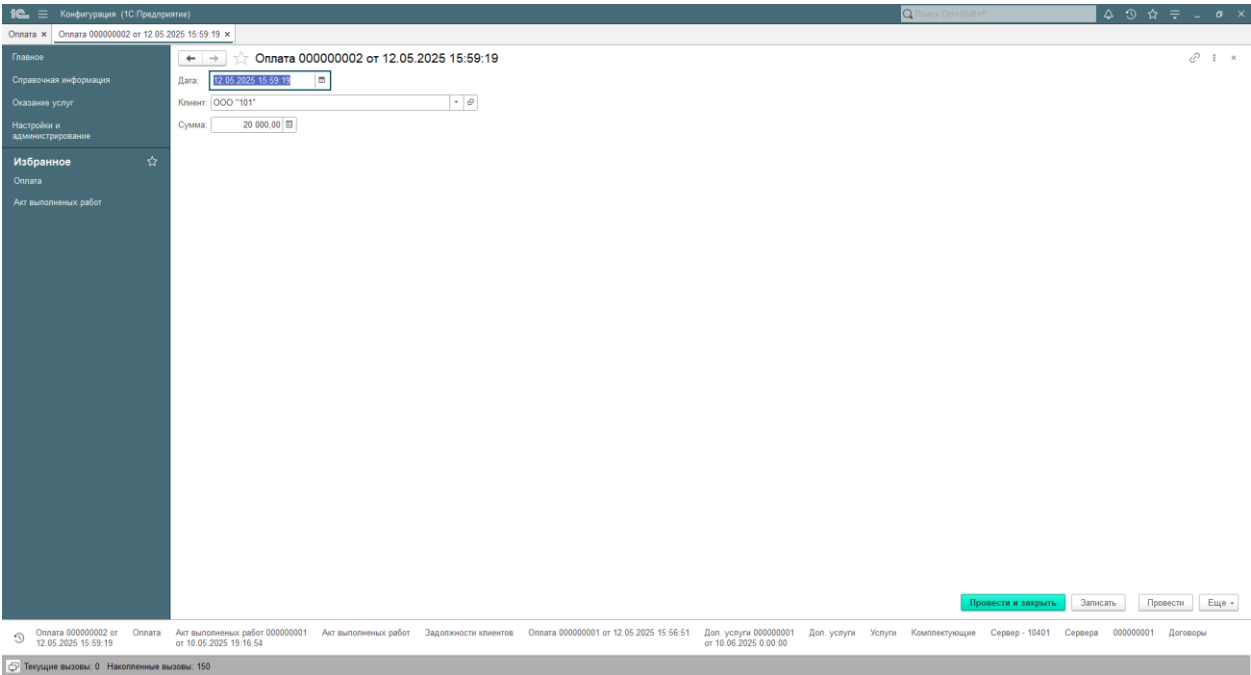


Рисунок В.5 – Создание документа «Оплата»

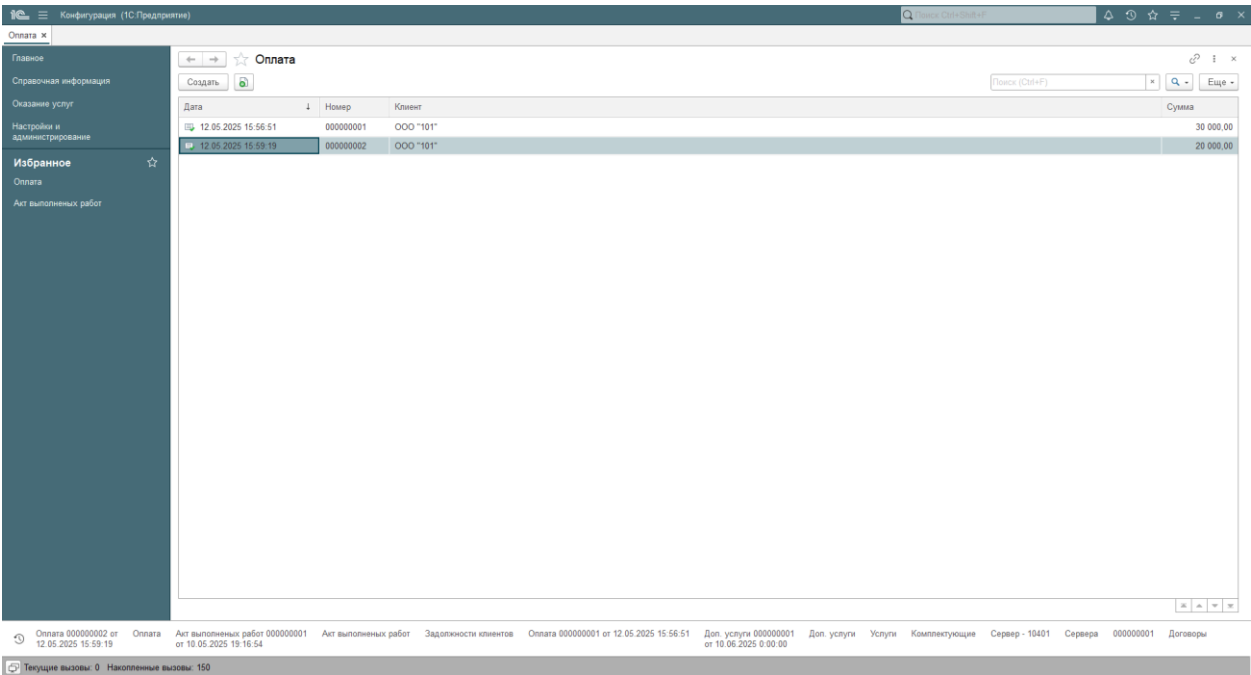


Рисунок В.6 – Форма списка документов «Оплата»

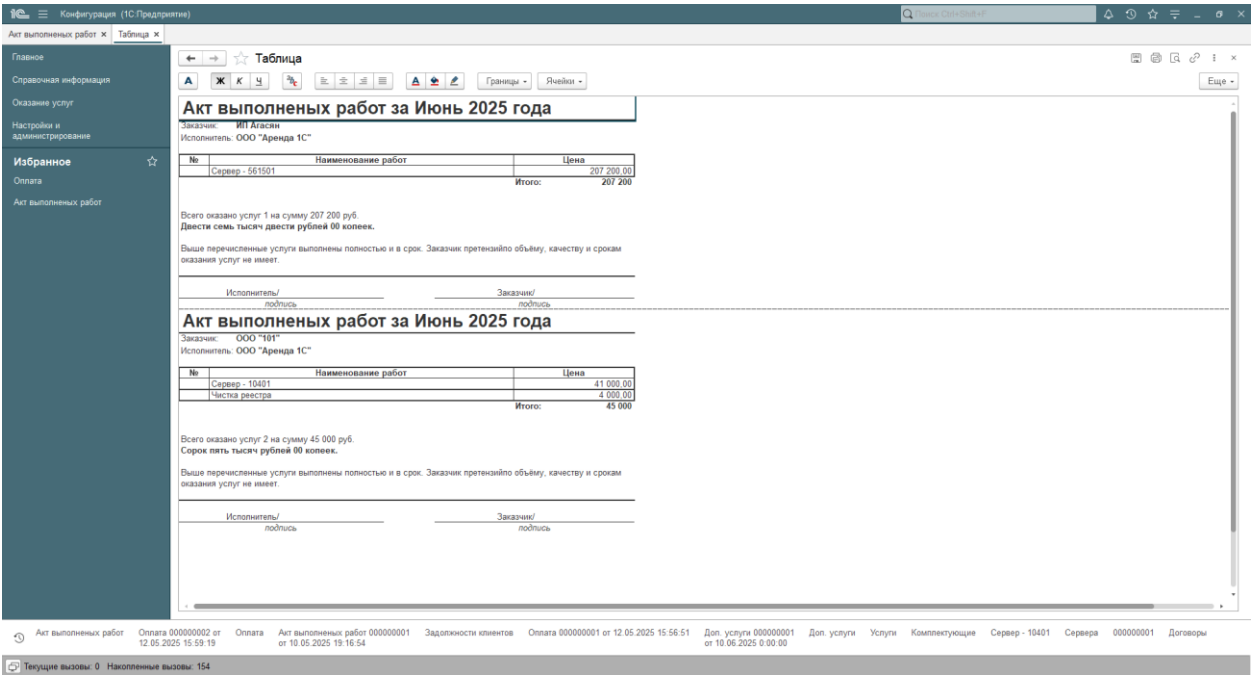


Рисунок В.7 – Формирование печатной формы документа «Акт выполненных работ»

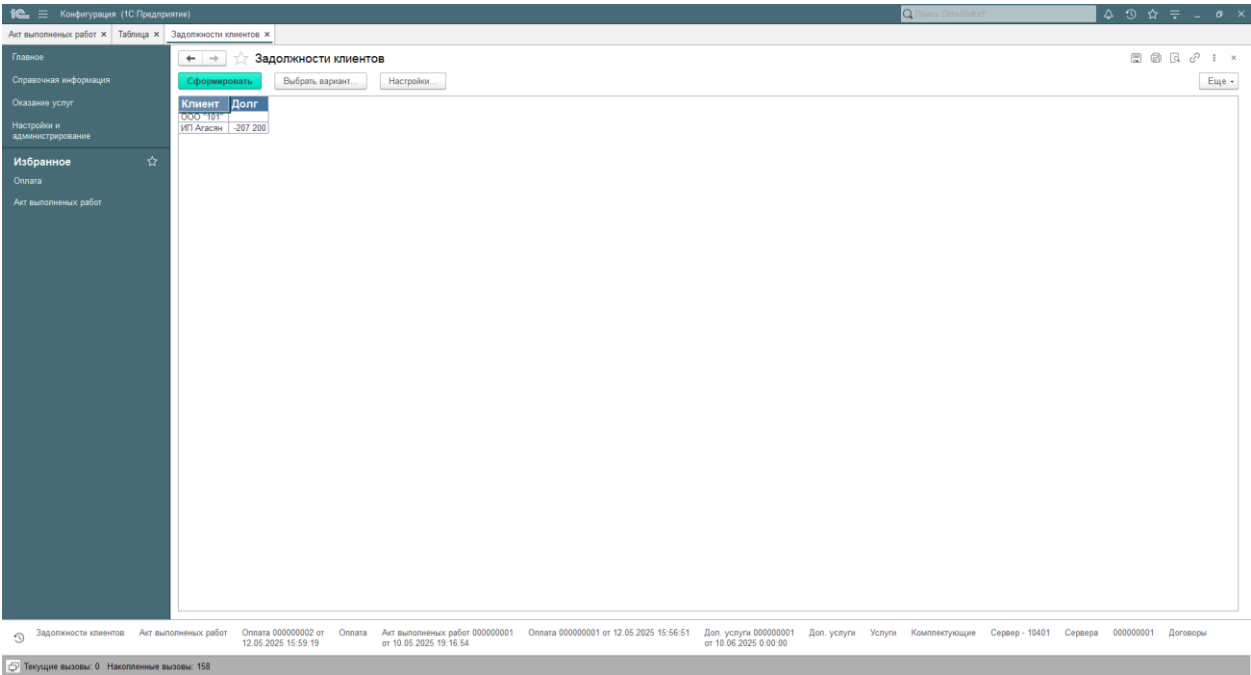


Рисунок В.8 – Формирование отчёта «Задолжности клиентов»