

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Технологический институт –
филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»
(ТИ НИЯУ МИФИ)

Кафедра информационных технологий и прикладной математики

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ
РАБОТА БАКАЛАВРА**

направление подготовки 09.03.01 «Информатика и вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Тема: Разработка программного модуля для формирования реестра
договоров, подлежащих удалению из баз данных после завершения
сроков хранения

Руководитель
Начальник отдела в
подразделении

(должность)

(подпись)

А.А. Дорофеев
(И.О. Фамилия)

Студент
ИВТ-41Д

(группа)

(подпись)

В.А. Телеш
(И.О. Фамилия)

ВКР допущена к защите в ГЭК
Заведующий кафедрой

Н.В. Чупракова
«» июня 2025 г.

ВКР защищена « » июня 2025 г.

Протокол ГЭК № _____ на оценку « _____ »

Секретарь ГЭК _____ /Ю.А. Порохина /

г. Лесной – 2025 г.

СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ	3
ВВЕДЕНИЕ	4
РАЗДЕЛ 1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ	7
1.1 Анализ предметной области.....	7
1.2 Описание бизнес-процесса.....	12
1.3 Средства разработки программного обеспечения	18
РАЗДЕЛ 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ	23
2.1 Проектирование базы данных.....	23
2.2 Разработка серверной части приложения	28
2.3 Проектирование клиентской части.....	36
РАЗДЕЛ 3 ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЯ	40
3.1 Руководство пользователя	40
3.2 Планируемый эффект от внедрения	56
ЗАКЛЮЧЕНИЕ	59
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	60
ПРИЛОЖЕНИЕ А	62
ПРИЛОЖЕНИЕ Б.....	66
ПРИЛОЖЕНИЕ В	69
ПРИЛОЖЕНИЕ Г.....	71
ПРИЛОЖЕНИЕ Д	72
ПРИЛОЖЕНИЕ Е.....	73

СПИСОК СОКРАЩЕНИЙ

АСКИОП – автоматическая система контроля и исполнения оперативных поручений;

СЭД – система электронного документооборота;

Задача WC03P – единая система электронного документооборота;

Задача WC07P – единая система электронного согласования документов;

Задача MC49P – система управления договорной деятельностью.

ВВЕДЕНИЕ

На данный момент информационные технологии представляют собой одно из ключевых направлений, определяющих развитие мировой экономики. Многие организации сталкиваются с необходимостью эффективного управления информацией, соответственно возникает вопрос ее хранения и удаления. К ключевым задачам этой области относится соблюдение нормативных требований по срокам хранения документов, что в особенности актуально для учреждений, работающих с большим объемом данных.

ФГУП «Комбинат «Электрохимприбор» входит в число ведущих подведомственных предприятий Государственной корпорации «Росатом». В рамках своей деятельности предприятие постоянно стремится к совершенствованию производственных процессов, внедрению инновационных технологий и повышению эффективности работы. В современном мире, где объемы информации растут с каждым годом, особое значение приобретает эффективное управление большими массивами данных. В связи с этим на предприятии активно внедряются передовые методы и современные информационные технологии, позволяющие оптимизировать процессы обработки, хранения и анализа информации.

С учетом вышеуказанных факторов было принято решение разработать программный модуль для формирования реестра договоров, подлежащих удалению из баз данных после завершения сроков хранения. Для реализации данной задачи планируется использовать современные технологии веб-программирования, что обеспечит удобство и гибкость в управлении процессом. Такой подход позволит автоматизировать процесс формирования реестра и повысить эффективность работы. Также поможет в оптимизации объема хранимой информации и экономии дорогостоящих ресурсов системы хранения данных комбината. В результате предприятие сможет более эффективно использовать свои ресурсы и снизить операционные издержки за счет своевременного удаления устаревших данных.

Актуальность выпускной квалификационной работы заключается в необходимости перехода на импортонезависимое программное обеспечение. В настоящее время все важные данные и информационные ресурсы хранятся в базе данных, реализованной на сервере Microsoft SQL Server (MSSQL) [2]. Однако, в связи с изменением внешних условий, требованиями к независимости от иностранных решений и стратегией оптимизации инфраструктуры, возникает необходимость миграции данных на альтернативную платформу, такую как Postgres Pro. Использование веб-технологий при создании программного модуля значительно упростит процесс интеграции его с новым сервером. Кроме того, веб-ориентированный подход позволяет обеспечить доступ к данным через браузеры, что повышает удобство работы пользователей и расширяет функциональные возможности системы.

Объектом исследования выпускной квалификационной работы является процесс задачи МС49Р «Автоматизированная система управления договорной деятельностью».

Предметом исследования в рамках данной работы является процесс удаления скан-копий приложенных документов к договорам, которые имеют отметки «Договор закрыт» или «Договор исполнен». В частности, рассматривается процедура автоматического удаления файлов согласования разовых сделок, заявок на платежи, актов закрытия договоров и других связанных с ними документов по истечении пяти лет после окончания календарного года, в котором был закрыт или исполнен соответствующий договор или договорной документ. Такой подход позволяет обеспечить соблюдение нормативных требований по срокам хранения информации, а также снизить объем хранимых данных за счет своевременного их удаления.

Целью данной работы является разработка программного модуля для анализа и визуализации данных о договорах, направленных на закрытие. Кроме того, модуль будет обеспечивать формирование реестра договоров, которые подлежат удалению из баз данных по истечении установленных

сроков хранения. Данная программа предоставит возможность выполнять следующие функции:

- просмотр договоров, направленных на закрытие;
- просмотр закрытых договоров и прикрепленных к ним файлам;
- составление реестра для последующего удаления закрепленных за договорами файлов;
- удаление связей между договорами и закрепленными за ними файлами в базе данных.

Приложение должно разрабатываться на языке программирования Python с использованием фреймворков Django и Django Rest Framework. Веб страницы должны быть разработаны с помощью языка разметки HTML, языка таблиц стилей CSS, языка программирования JavaScript и библиотеки Bootstrap5.

Для реализации поставленной цели необходимо решить следующие задачи:

1. Провести анализ предметной области;
2. Изучить существующую на предприятии информационную систему для дальнейшей интеграции;
3. Ознакомиться с базой данных;
4. Дополнить существующую базу данных новыми структурами для расширения существующего функционала;
5. Разработать серверную часть приложения;
6. Спроектировать клиентскую часть приложения;
7. Выполнить тестирование программного продукта.

РАЗДЕЛ 1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ

1.1 Анализ предметной области

Согласно распоряжению об удалении информации, для экономии дорогостоящих ресурсов системы хранения данных комбината, необходимо оптимизировать объем хранимой информации [8]. В задаче МС49Р необходимо удалить все скан-копии приложенных документов к договорам с отметкой «Договор закрыт» или «Договор исполнен» по истечении пяти лет после окончания года закрытия договора/исполнения договорного документа.

В настоящий момент пятнадцатого числа каждого месяца срабатывает процедура определения закрытых договоров. Закрытым считается договор, период действия которого завершен. В случае, если обязательства по договору не исполнены, система позволяет продлить время исполнения, но для подразделения-исполнителя необходимо сформировать дополнительное соглашение. Если же обязательства исполнены, система направляет исполнителю извещение по WC03P с просьбой запустить процедуру закрытия электронного договора. Если исполнитель не запустил процедуру закрытия в течение месяца, то следующего пятнадцатого числа, процедура вновь срабатывает и направляет извещение. Через три месяца с первого извещения, процедура проверяет, что никаких действий с договором не производилось и формирует поручение в системе АСКИОП, которое также дает месяц на исполнение процедуры закрытия. Если договор по-прежнему не закрыт, то начальник подразделения-исполнителя получает снижение индивидуальной стимулирующей надбавки.

В настоящее время для отслеживания договоров, которые находятся на стадии закрытия, используется система, основанная на процедурах, хранящихся в базе данных MSSQL. Однако, в рамках стратегического плана по минимизации санкционных рисков и повышению информационной безопасности, в будущем предполагается осуществить переход всех данных на новую платформу, а именно на систему управления базами данных (СУБД)

Postgres PRO. Этот переход позволит не только улучшить производительность и надежность системы, но и значительно упростит конвертацию данных в веб-формат.

На предприятии «ФГУП Комбинат Электрохимприбор» существуют отделы, которым важно просматривать и анализировать исполнение договоров. Необходимо реализовать веб-интерфейс с возможностью просмотра сформированных за конкретный промежуток времени писем, которые либо уже были направлены в АСКИОП, либо ждут скорого направления.

Также необходимо сформировать реестр договоров, подлежащих удалению из баз данных, после завершения их сроков хранения. Веб-приложение должно иметь интерфейс, позволяющий взаимодействовать с реестром, что включает занесение и удаление договоров, а также просмотр прикрепленных к ним файлов.

Для реализации данных задач необходимо разработать программный модуль с использованием имеющейся базы данных, чтобы в дальнейшем, при переходе основной системы на веб-формат, было проще интегрировать функционал нового приложения с основным. Для разработки были предоставлены базы данных «ДОГОВОР», «FileStore» и «WC03P».

База данных «ДОГОВОР» содержит в себе всю необходимую информацию по договорам. Список использованных из нее таблиц и представлений предоставлен в таблице 1.1.

Таблица 1.1 – Использованные таблицы и представления БД «ДОГОВОР»

№	Имя	Тип
1	_договор_банковская_гарантия	Представление
2	_договор_банковская_гарантия_WC07P	Представление
3	_договор_источники_финансирования	Представление
4	_договор_условия_расчетов_3	Представление
5	_договор_условия_расчетов_4	Представление

№	Имя	Тип
6	_договор_условия_расчетов_авансирование	Представление
7	_договор_условия_расчетов_закрытие_аванса	Представление
8	_договор_доп_соглашение	Представление
9	_предприятие	Представление
10	Журнал_отправки_писем_по_закрытию_договора	Таблица

База данных управления договорной деятельностью является важным инструментом для получения достоверной и точной информации о договорах. В данном веб-приложении будут использоваться таблицы, в которых содержатся основные данные, необходимые для решения предоставленных задач.

Для получения детальной информации о договорах были использованы уже существующие в базе данных представления, такие как:

- «_договор_доп_соглашение» – комплексное представление данных о дополнительных соглашениях к договорам. Оно служит инструментом для хранения расширенной информации о договорах. Содержит даты начала и окончания действия договоров и соглашений, условия оплаты, поставки. Включает суммы по спецификациям без НДС, с НДС, в рублях, что важно для финансового учета. Хранит номера и даты регистраций документов, статусы договоров, сведения о госконтрактах. Содержит сведения о закупках, графиках поставок, исполнении обязательств.
- «_предприятие» – данное представление служит инструментом для хранения, анализа и управления информацией о юридических лицах или организациях, участвующих в бизнес-процессах, государственных закупках, контрактах и других взаимодействиях. Хранит основные реквизиты предприятий: ИНН, КПП, ОГРН, полное и краткое наименование. Включает данные о регионе, городе, адресах (улица/дом), почтовом индексе. Содержит сведения о подразделениях и

пользователях, связанных с предприятием, а также юридические и налоговые параметры.

- «_договор_источники_финансирования» – представление содержит информацию для осуществления учета, анализа и контроля источников финансирования в рамках контрактных или проектных систем. Позволяет отслеживать суммы выделенного финансирования, коды источников и другие параметры.
- «_договор_банковская_гарантия_WC07P» – представление предназначено для хранения и управления информацией о банковских гарантиях, связанных с контрактами или сделками в рамках единой системы электронного согласования документов. Хранит сведения о различных видах гарантий, предоставляемых по контрактам. Содержит информацию о сроках действия, типе обеспечения и другие параметры.
- «_договор_банковская_гарантия» – данное представление используется для учета всех видов гарантий, их характеристик, сроков действия, связанных документов и участников. Хранит сведения о предоставленных гарантиях, включая тип и вид обеспечения. Включает номер документа, дату его оформления и сумму обеспечения.
- «_договор_условия_расчетов_авансирование» – представление содержит информацию о процентах аванса, сроках его оплаты и условиях, связанных с предоплатой.
- «_договор_условия_расчетов_закрытие_аванса» – представление хранит информацию об осуществлении закрытия авансовых платежей по контрактам.
- «_договор_условия_расчетов_3» – представление предназначено для хранения информации о процентных ставках расчетов, сроках оплаты и условиях, связанных с расчетами по договору.

- «_договор_условия_расчетов_4» – данное представление содержит в себе информацию о сроках поставки, условиях и датах, связанных с логистикой по договору.

Таблица «Журнал_отправки_писем_по_закрытию_договора» включает в свой состав атрибут «документ_ID_договора», который выполняет функцию связующего ключа, с базой данных единой системы электронного документооборота «WC03P». Структура данной сущности приведена в таблице 1.2.

Таблица 1.2 – Список полей таблицы

«Журнал_отправки_писем_по_закрытию_договора»

№	Имя	Тип данных	Описание
1	RID	serial	Уникальный идентификатор
2	предприятие_ID	integer	Идентификатор предприятия
3	договор_ID_основной	varchar	Идентификатор договора
4	подразделение_ведущее_договор	varchar	Подразделение ведущее договор
5	куратор_договора	varchar	Куратор договора
6	количество_отправленных_писем	smallint	Количество отправленных писем
7	дата_отправки_письма	datetime	Дата отправки письма
8	документ_ID_сэд	varchar	Идентификатор документа СЭД
9	адресат_ID_xml	varchar	Идентификатор адресата
10	text_doc	varchar	Текст документа

База данных «FileStore» предназначена для хранения и управления информацией о файлах. В рамках данной базы данных содержится подробная информация о каждом файле, закрепленном за соответствующим договором,

включающая такие параметры, как уникальный идентификатор файла, его наименование, объем и структура архива, в которой представлен данный файл. Для удобства работы с данными о файлах в базе данных реализовано специальное представление под названием «_файл_карточка». Для удаления связей между файлами реализована процедура «_файл_задача_связь_del», код которой представлен в приложении Б.1. Структура представления «_файл_карточка» описана в таблице 1.3.

Таблица 1.3 – Список полей представления «_файл_карточка»

№	Имя	Тип данных	Описание
1	файл_ID	bigint	идентификатор файла
2	структура_архива	varchar	структура архива
3	имя_файла	varchar	наименование файла
4	суммарный_размер	varchar	размер файла

База данных «WC03P» предназначена для централизованного хранения и эффективного управления информацией, связанной с единой системой электронного документооборота (СЭД). Основная функция базы данных «WC03P», в рамках данного программного модуля, заключается в поддержке процедур получения информации о переписке, связанной с конкретными документами внутри системы электронного документооборота. Это позволяет пользователям получать полную картину коммуникаций по конкретным договорам.

1.2 Описание бизнес-процесса

Для решения бизнес-задачи предлагается спроектировать информационную систему для определенного отдела, либо нескольких отделов предприятия, отдельную функциональную подсистему или группу задач на основе программного продукта и процессного подхода. Для этого необходимо [12]:

- составитель бизнес-модели процессов или функционирования ресурсов в контексте архитектуры предприятия и их автоматизации в ФГУП «Комбинат «Электрохимприбор»;
- обосновать выбор программного продукта и требований к его модификациям, проанализировать соответствия функционала по задаче и моделям бизнес-процессов.

На данном предприятии при моделировании используются следующие типы диаграмм:

- диаграмма цепочки добавленной стоимости (VAD – Value Added Chain Diagram) помогает в описании взаимодействия процессов верхнего уровня;
- диаграмма плавательных дорожек (SLD – Swim Lane Diagram) используется для описания процессов среднего и нижнего уровней;
- диаграмма расширенной цепочки процессов, управляемой событиями, (Eepc – Extended Event Driven Process Chain) применяется при описании процессов верхнего, среднего и нижнего уровней;
- диаграмма организационной структуры (OC – Organizational Chart) применяется для описания организационных моделей;
- диаграмма дерева функций (FT – Function Tree) применяется при описании структуры процессов и функций;
- диаграмма типов информационных систем (ASTD – Application System Type Diagram) применяется для описания структуры информационных систем.

Для осуществления автоматизации данного процесса необходимо достичь понимания структуры его разработки. Модель бизнес-процесса предоставляет полную схему процесса и позволяет определить конкретные компоненты, которые подлежат автоматизации при помощи программного обеспечения.

Диаграмма внешнего уровня бизнес-процесса демонстрирует весь процесс в целом, без углубления в детали его выполнения. Она показывает общую картину процесса и выделяет ключевые этапы, которые обычно охватываются в процессе. Примерами могут быть этапы, связанные со сбором данных, анализом информации, разработкой планов и стратегий, выпуском продукта или услуги. Также может отображать, как различные участники взаимодействуют между собой в рамках процесса, и показывает, какие ресурсы требуются для успешного выполнения процесса. Во многих случаях может быть использована для быстрой оценки процесса и его ключевых характеристик.

Таким образом, диаграмма внешнего уровня бизнес-процесса отображает общую картину процесса и его этапы, участников, ресурсы и взаимодействия между ними. Она является полезным инструментом для понимания процесса и его ключевых характеристик, но без раскрытия подробностей и мелких деталей, которые могут быть выявлены на углубленных уровнях диаграммы [6].

Диаграмма внутреннего уровня бизнес-процесса – это более детальная диаграмма, чем диаграмма внешнего уровня, которая позволяет представить процесс в более подробной форме. В отличие от диаграммы внешнего уровня, диаграмма внутреннего уровня описывает каждый этап процесса и его потоки, включая каждый шаг и взаимодействие между участниками. Эта диаграмма может быть использована для детального анализа процесса, выявления недостатков и определения возможных решений.

Благодаря диаграмме внутреннего уровня можно обнаружить узкие места в процессе, найти возможности для его улучшения и автоматизации. Она также может быть использована для обучения новых сотрудников и документирования текущих рабочих процедур.

Таким образом, диаграмма внутреннего уровня бизнес-процесса позволяет представить детальный и подробный обзор процесса, его элементов и взаимодействия между ними. Она может использоваться для анализа и

оптимизации процесса в рамках компании, что способствует улучшению производительности и повышает удовлетворенность клиентов [1].

В моделировании бизнес-процесса была использована диаграмма swim-lane по следующим причинам:

- позволяет визуально отобразить бизнес-процесс с большим количеством участников;
- позволяет структурировать сложные бизнес-процессы, повышая удобство в использовании для обсуждения процесса с коллегами и партнерами;
- позволяет упростить описание проблемных моментов и промежуточных результатов для каждого этапа;
- как приложение к другой нотации на основе условных графиков может повысить ее удобство использования для пользователя благодаря определению более детальных связей между разными процессами.

Таким образом, использование нотации «swim-lane» упрощает понимание бизнес-процессов и дает полное представление о действиях и ответственности в необходимой последовательности. В рамках текущей системы закрытия договоров выделяются два взаимосвязанных бизнес-процесса, каждый из которых играет ключевую роль в обеспечении эффективного функционирования всей системы. Первый бизнес-процесс, связанный с заведением и направлением договоров на закрытие, представлен на рисунке 1.1. Второй бизнес-процесс является продолжением первого, охватывает этапы удаления информации по закрытым договорам из базы данных и представлен на рисунке 1.2.

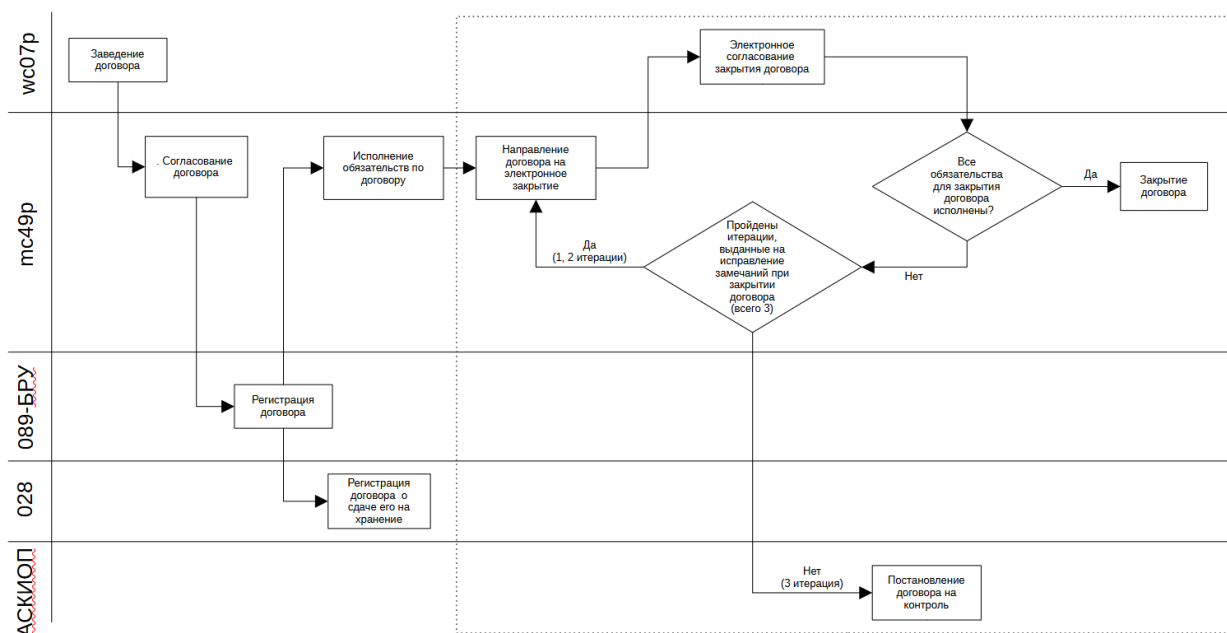


Рисунок 1.1 – Внешний уровень первого бизнес-процесса

Процесс управления договорами начинается с их создания, осуществляемого сотрудниками отделов 055, 077 или 079. На следующем этапе, в рамках задачи MC49P, осуществляется согласование договора, в котором принимают участие сотрудники ранее указанных отделов, а также сотрудники отдела 057. После успешного согласования договор передается на регистрацию в подразделение 028, ответственное за его хранение.

По завершении всех обязательств, изложенных в договоре, он переходит к процедуре электронного закрытия. Согласование данного этапа осуществляется в рамках задачи WC07P, где проверяется выполнение всех условий и обязательств, предусмотренных договором. В случае, если все обязательства были выполнены, договор подлежит закрытию. Однако, если какие-либо обязательства не были выполнены, формируется дополнительное соглашение, направленное на устранение замечаний, возникших в процессе закрытия договора. На выполнение условий данного соглашения устанавливается срок в один месяц.

Если дополнительное соглашение о выполнении обязательств было сформировано три раза, но так и не было реализовано, договор ставится на контроль в системе АСКИОП, что позволяет обеспечить дополнительный

уровень мониторинга и управления рисками, связанными с невыполнением условий договора. Таким образом, данный процесс способствует повышению прозрачности и ответственности на всех этапах их жизненного цикла.

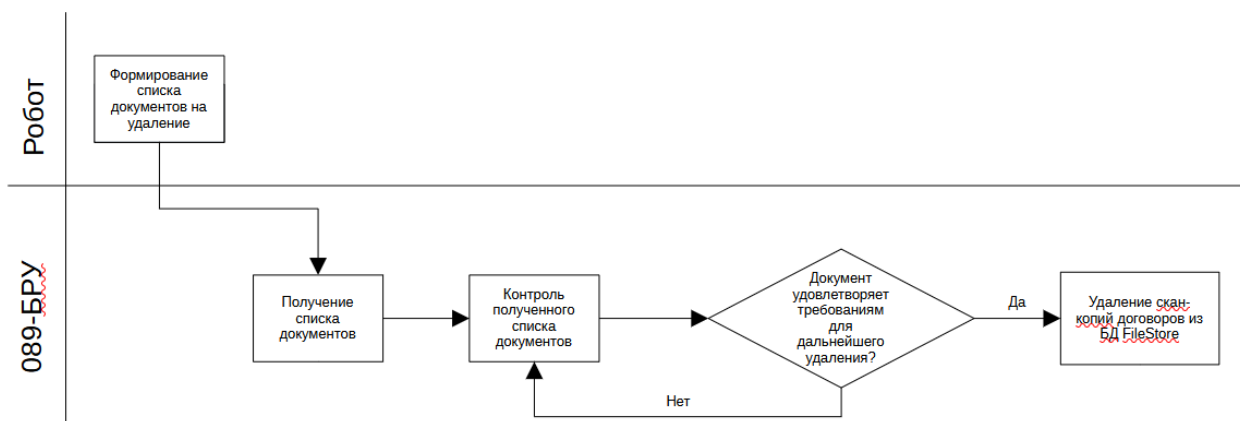


Рисунок 1.2 – Внешний уровень второго бизнес-процесса

Процесс управления договорами завершается этапом, на котором успешно закрытые договоры переходят к следующему бизнес-процессу. По истечении установленного срока хранения договора в базе данных, автоматизированная система инициирует формирование списка документов, подлежащих удалению. Этот процесс является важной частью управления жизненным циклом документов и обеспечивает соблюдение норм и стандартов, касающихся хранения и утилизации информации.

Сформированный список документов подлежит внимательному рассмотрению сотрудниками отдела 089. В ходе этой проверки осуществляется оценка соответствия каждого документа установленным критериям, необходимым для дальнейшего удаления. В случае, если документ соответствует всем требованиям, связанным с его удалением, из базы данных «FileStore» производится удаление всех скан-копий, относящихся к данному договору. Иначе производится дополнительная проверка договора и приложенных к нему документов.

Удаление устаревших или ненужных документов позволяет минимизировать риски, связанные с хранением избыточной информации, а

также способствует повышению общей эффективности работы организации. Таким образом, реализация данного этапа бизнес-процесса является важным шагом в направлении улучшения управления документами и повышения прозрачности всех операций, связанных с жизненным циклом договоров.

1.3 Средства разработки программного обеспечения

Техническое задание содержит подробное описание конкретного стека технологий, применяемого для создания веб-сервисов, который включает такие компоненты, как язык разработки и язык разметки.

Серверная часть была реализована с использованием следующего набора веб-технологий:

- Python – это высокоуровневый язык программирования общего назначения, который обладает динамической строгой типизацией и автоматически управляет памятью. Он направлен на повышение производительности разработчика, улучшение читаемости кода и его качества, а также на обеспечение переносимости программ, написанных на нем. Язык полностью объектно-ориентированный, что означает, что все его элементы являются объектами [19].
- Django – это фреймворк для веб-приложений на языке Python, позволяющий быстро создавать безопасные и поддерживаемые веб-сайты, использующий шаблон проектирования MVC.
- MVC («Model-View-Controller», «Модель-Вид-Контроллер») – схема, которая разделяет данные приложения и бизнес-логику на три отдельных компонента, позволяет модифицировать каждый из них независимо. В Django этот подход реализуется через паттерн Model-View-Template (MVT), где модель (Model) отвечает за управление данными, представление (View) отображает результаты пользовательских действий, а шаблоны (Template) используются для визуального оформления вывода на сайте.

Таким образом, в Django Model представляет базу данных, а View и Template предоставляют пользовательский интерфейс.

MVT-Pattern в Django позволяет разработчикам создавать приложения, разделённые на отдельные компоненты, что приводит к созданию модульных приложений, которые легче поддерживать другим специалистам. Это также упрощает процесс улучшения и адаптации приложения к конкретным задачам.

- Django Rest Framework (DRF) – библиотека, работающая со стандартными моделями Django для создания гибкого и мощного API.
- API (Application Programming Interface) – описание способов взаимодействия одной компьютерной программы с другими.
- Postgres Pro – реляционная система управления базами данных (СУБД).
- HTTP (Hypertext Transfer Protocol) – сетевой протокол прикладного уровня для передачи данных между клиентом и сервером. Данный протокол определяет формат и правила передачи данных в процессе сетевого взаимодействия между персональными компьютерами. В основе работы протокола лежат запросы и ответы с запрошенной информацией, то есть клиент отправляет запрос на сервер, сервер в свою очередь возвращает ответ с запрошенной информацией. Для отправки запросов используются методы: GET, POST, PUT и прочие. Каждый метод определяет тип действия, которое необходимо выполнить на сервере.

Клиентская часть была реализована при помощи следующих инструментов веб-разработки:

- HTML (HyperText Markup Language) – это стандартизированный язык разметки документов, применяемый для создания веб-страниц и определения их структуры и содержания. Он состоит из набора тегов и атрибутов, используемых для описания контента и форматирования веб-страницы.
- CSS (Cascading Style Sheets) – формальный язык описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки.

- JavaScript – язык программирования, используемый для создания интерактивных веб-страниц и приложений. Позволяет добавлять функциональность на веб-страницы. Используется как встраиваемый язык для программного доступа к объектам приложений [13].
- Bootstrap – набор инструментов для ускоренной верстки сайтов и веб-приложений. Включает в себя HTML и CSS шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript расширения [14].

Для применения этого стека был выбран редактор кода Visual Studio Code. С его помощью можно быстро создавать проекты, он подсвечивает синтаксис кода и помогает автоматически исправлять ошибки, а также предлагает широкий набор функций и инструментов. Среда разработки представлена на рисунке 1.3.

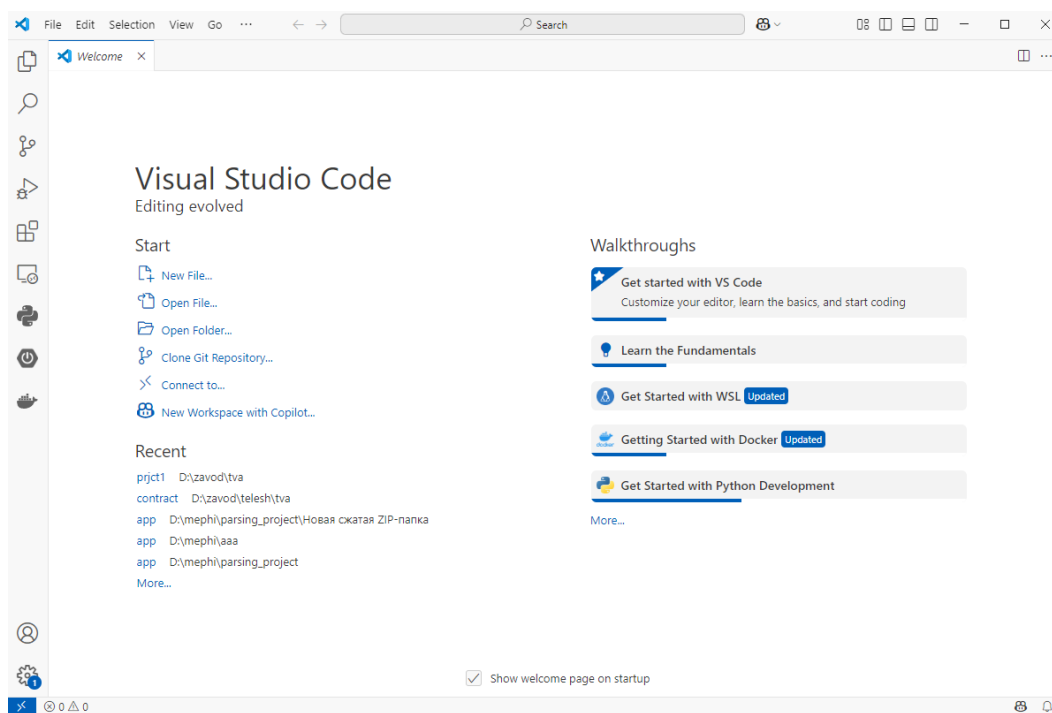


Рисунок 1.3 – Редактор кода Visual Studio Code

Управление СУБД осуществляется с использованием графической среды разработки и администрирования dbForge Studio for MySQL. Визуальное представление среды системы управления базами данных представлено на рисунке 1.4.

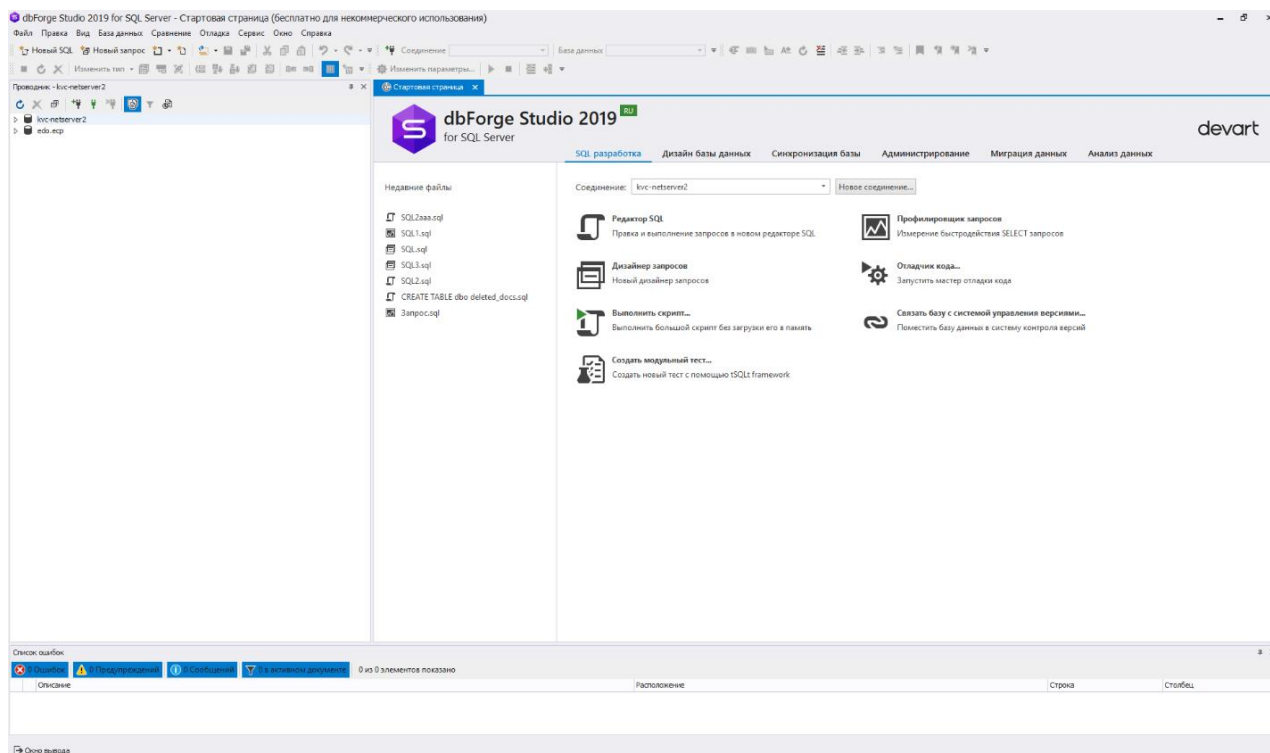


Рисунок 1.4 – dbForge Studio for MySQL

Получение данных о договорах осуществляется посредством реализации интерфейса REST API, разработанного на базе современного фреймворка Django REST Framework. Данный API обеспечивает стандартизированный и безопасный механизм обмена информацией между серверной частью системы и внешними клиентскими приложениями, что позволяет осуществлять эффективное получение, обновление и управление данными о договорах в автоматизированной среде. На рисунке 1.5 приведен перечень ссылок, используемых для работы программного модуля.

```
Using the URLconf defined in main.urls, Django tried these URL patterns, in this order:
1. admin/
2. [name='index']
3. close/
4. doc-card/<str:heading_name>/ [name='doc-card']
5. api/ dlist/
6. api/ ddlist/
7. api/ cma/
8. api/ show-files/
9. api/ show-added-files/
10. api/ send-doc-id-sed/
11. api/ get-msgs/
12. api/ del-select-links/
13. api/ checked-add/
14. api/ checked-del/
15. api/ del-table-filter/
16. api/ add-table-filter/
17. api/ del-link/
18. api/ no-contract-list/
```

Рисунок 1.5 – DjangoRestFramework для работы с API

Вывод: в соответствии с техническим заданием на разработку программного модуля, в данном разделе были выполнены следующие задачи: проведен анализ предметной области, в ходе которого были определены термины, используемые в информационной системе; построены модели бизнес-процесса; а также описаны инструментальные средства, необходимые для реализации веб-сервиса.

РАЗДЕЛ 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ

2.1 Проектирование базы данных

Для реализации реестра договоров, подлежащих удалению после завершения сроков хранения, в базе данных «ДОГОВОР» была создана таблица «deleted_docs», структура которой представлена в таблице 2.1. Реестр хранит в себе информацию о скан-копиях приложенных к договорам документов.

Таблица 2.1 – Список полей таблицы «deleted_docs»

№	Имя	Тип данных	Описание
1	dels_contract_id_main	varchar	Идентификатор договора
2	dels_file_id	bigint	Идентификатор файла
3	dels_doc_id	bigint	Идентификатор документа
4	dels_file_size	bigint	Размер файла
5	dels_date_del	datetime	Дата удаления
6	dels_date_close	datetime	Дата закрытия
7	dels_date_create	datetime	Дата заведения
8	dels_code_access	varchar	Код доступа
9	dels_file_name	varchar	Имя файла

Для получения актуальных сведений о договорах, срок хранения которых истек, было реализовано представление «_del_docs». Данное представление предназначено для автоматизированного извлечения информации о договорах, по которым прошло не менее пяти лет с момента завершения календарного года их закрытия или исполнения. Оно позволяет получать данные о связанных с ними скан-копиях и иных приложенных документах, для последующей загрузки этой информации в реестр. Структура данного представления отражена в таблице 2.2. В качестве примера реализации данного функционала приведен соответствующий код в Приложении А.

Таблица 2.2 – Список полей представления «_del_docs»

№	Имя	Тип данных	Описание
1	договор_ID_основной	varchar	Идентификатор договора
2	документ_ID	bigint	Идентификатор документа
3	файл_ID	bigint	Идентификатор файла
4	код_доступа	varchar	Код доступа к файлу
5	имя_файла	varchar	Имя файла
6	дата_закрытия	datetime	Дата закрытия

Для отображения на странице сведений о самих договорах, срок хранения которых истек, было реализовано представление «_del_docs_list_of_contracts». Структура данного представления отражена в таблице 2.3. В качестве примера реализации данного функционала приведен соответствующий код в Приложении А.

Таблица 2.3 – Список полей представления «_del_docs_list_of_contracts»

№	Имя	Тип данных	Описание
1	договор_ID_основной	varchar	Идентификатор договора
2	номер_договора	varchar	Номер договора
3	дата_договора	datetime	Дата договора
4	город_поставщик	varchar	Город поставщик
5	кто_платит	varchar	Кто платит
6	сумма_договора_с_ндс	numeric	Сумма договора с НДС
7	валюта_договора	varchar	Валюта договора
8	дата_закрытия	datetime	Дата закрытия
9	регистрация_089	varchar	Регистрация в 089 отделе

Для обеспечения возможности отображения на пользовательском интерфейсе информации о договорах, срок хранения которых истек и которые уже были внесены в реестр, было реализовано представление «_deleted_docs_list_of_contracts». Структура данного представления отражена

в таблице 2.4. В качестве примера реализации данного функционала приведен соответствующий код в приложении А.

Таблица 2.4 – Список полей представления «_deleted_docs_list_of_contracts»

№	Имя	Тип данных	Описание
1	договор_ID_основной	varchar	Идентификатор договора
2	номер_договора	varchar	Номер договора
3	дата_договора	datetime	Дата договора
4	город_поставщик	varchar	Город поставщик
5	кто_платит	varchar	Кто платит
6	сумма_договора_с_ндс	numeric	Сумма договора с НДС
7	валюта_договора	varchar	Валюта договора
8	dels_date_close	datetime	Дата закрытия
9	регистрация_089	varchar	Регистрация в 089 отделе
10	is_dels_date_del	boolean	Флаг удаления связи

Для отображения на странице информации о договорах, направленных на закрытие, было реализовано представление _list_to_closing. Структура данного представления отражена в таблице 2.5. В качестве примера реализации данного функционала приведен соответствующий код в приложении А.

Таблица 2.5 – Список полей представления _list_to_closing

№	Имя	Тип данных	Описание
1	договор_ID_основной	varchar	Идентификатор договора
2	документ_ID	bigint	Идентификатор документа
3	статус_договора	varchar	Статус договора
4	кто_платит	varchar	Кто платит
5	номер_договора	varchar	Номер договора
6	дата_договора	datetime	Дата договора
7	дата_документа	datetime	Дата документа

№	Имя	Тип данных	Описание
8	исполнитель_договора	varchar	Исполнитель договора
9	город_поставщик	varchar	Город поставщик
10	валюта_договора	varchar	Валюта договора
11	сумма_договора_без_ндс	numeric	Сумма договора без НДС
12	сумма_договора_ндс	numeric	Сумма НДС договора
13	сумма_договора_с_ндс	numeric	Сумма договора с НДС
14	РОСАТОМ_согласованная_с умма	numeric	Согласованная сумма с Росатомом
15	регистрация_089	varchar	Регистрация в 089 отделе
16	предприятие_ID	integer	Идентификатор предприятия
17	нарушение_сроков_поставк и	smallint	Количество нарушений сроков поставки
18	претензия_по_качеству	smallint	Претензии по качеству
19	эл_архив_договора	varchar	Наличие договора в электронном архиве
20	letters	integer	Количество отправленных писем
21	max_дата_отправки_письма	datetime	Последняя дата отправки письма

В рамках системы управления договорной деятельностью предусмотрена своевременная генерация уведомлений о необходимости направления договора на электронное закрытие. При этом, в единой системе электронного документооборота автоматически создается соответствующее извещение, информирующее ответственных лиц о необходимости выполнения процедуры закрытия договора. Для получения содержательной части данного уведомления была реализована хранимая процедура «mc49p_список_документов_для_закрытия_договоров», функционирующая в

базе данных «WC03P» [9]. В результате выполнения процедура выводит данные в виде таблицы, структура которой отражена в таблице 2.6. В качестве примера технической реализации данного функционала приведен соответствующий код в Приложении Б.

Таблица 2.6 – Список полей процедуры

«mc49p_список_документов_для_закрытия_договоров»

№	Имя	Тип данных	Описание
1	документ_ID_сэд	bigint	Идентификатор документа
2	номер_документа	varchar	Номер документа
3	дата_документа	datetime	Дата документа
4	текст_документа	varchar	Текст документа

В единой системе электронного документооборота для каждого отправленного извещения предусмотрена переписка, предназначенная для обсуждения деталей и уточнений, связанных с полученным уведомлением. Данная переписка позволяет участникам процесса своевременно обмениваться необходимой информацией по вопросам закрытия договорных обязательств. Для получения и отображения содержания данной переписки в базе данных «WC03P» была реализована хранимая процедура «_mc49p_список_переписка». Данная процедура предназначена для выборки всех сообщений, связанных с конкретным извещением. Полученные данные формируются в таблицу, структура которой представлена в таблице 2.7. В качестве примера технической реализации данного функционала в Приложении Б приведен соответствующий код.

Таблица 2.7 – Список полей процедуры _mc49p_список_переписка

№	Имя	Тип данных	Описание
1	автор	varchar	Автор сообщения
2	сообщение_дата	datetime	Дата сообщения
3	сообщение_текст	varchar	Текст сообщения
4	сообщение_общее	boolean	Флаг сообщения

№	Имя	Тип данных	Описание
5	документ_ID	bigint	Идентификатор документа
6	сообщение_ID	bigint	Идентификатор сообщения

2.2 Разработка серверной части приложения

В рамках разработки программного модуля был создан и настроен проект на базе фреймворка Django. На рисунке 2.1 представлена структурная схема созданного Django-проекта, которая иллюстрирует основные элементы его архитектуры, такие как настройки проекта, приложения, модели данных, маршруты и интерфейсы взаимодействия.

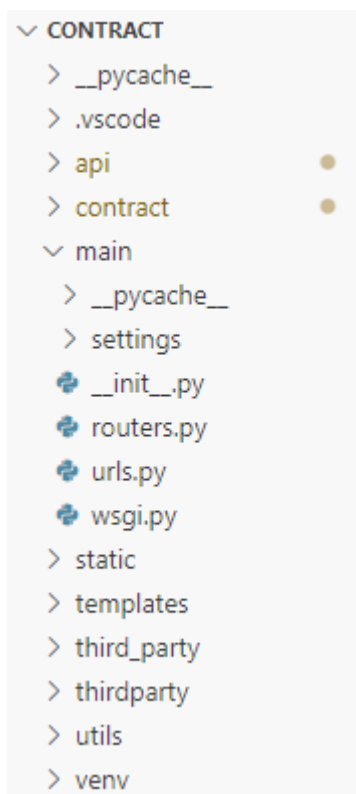


Рисунок 2.1 – Структура Django-проекта

Фреймворк Django предоставляет возможность создания самостоятельных модулей, известных как Django-приложения (apps), которые реализуют определённый функционал внутри проекта. Каждое Django-приложение может включать в себя модели данных, представления (views), шаблоны (templates), статические файлы, утилиты и другие компоненты, что

способствует организации кода по модульному и логическому принципу. Такой подход обеспечивает удобство разработки, поддержки и масштабирования системы, а также способствует разделению ответственности между различными функциональными блоками проекта [3].

В процессе разработки серверной части веб-сервиса были выполнены настройки конфигурационного файла, в частности, параметр `INSTALLED_APPS`, который определяет список приложений и библиотек, подключённых к проекту. В данный список были включены сторонние компоненты, такие как `rest_framework` — фреймворк для реализации RESTful API, а также `thirdparty.custom_filters` — сторонний модуль, предоставляющий расширенные возможности фильтрации данных [16]. Помимо этого, в состав проекта вошли собственные Django-приложения `contract` и `api`, предназначенные для реализации бизнес-логики и API-интерфейсов системы. Значение параметра `INSTALLED_APPS`, отражающее полный перечень подключённых модулей и библиотек, представлено в листинге 2.1 [20].

Листинг 2.1 – Значение параметра `INSTALLED_APPS`

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'contract.apps.ContractConfig',  
    'api.apps.ApiConfig',  
    'thirdparty.custom_filters',  
    'rest_framework'  
]
```

Статические файлы — это ресурсы, используемые для отображения веб-страницы, которые не изменяются в процессе эксплуатации сайта. К таким файлам относятся изображения, файлы стилей CSS, скрипты JavaScript и другие подобные ресурсы. Они обеспечивают визуальное оформление и функциональность интерфейса пользователя [11].

Фреймворк Django предоставляет встроенную систему управления статическими файлами, которая позволяет организовать их хранение, обработку и доставку клиенту. Настройка данной системы осуществляется в конфигурационном файле проекта. В основном, статические файлы размещаются в директории /static/, структура которой отражена на рисунке 2.2.

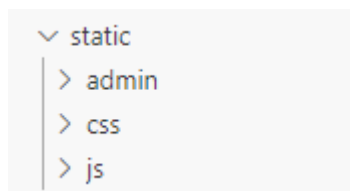


Рисунок 2.2 – Статические файлы

Для обеспечения корректной обработки статических файлов фреймворк Django требует указания путей поиска этих ресурсов. В частности, в файл конфигурации проекта была добавлена запись в параметр `STATICFILES_DIRS`, который определяет список директорий, где Django будет искать статические файлы проекта. Этот параметр позволяет гибко управлять расположением статических ресурсов и обеспечивает их правильную обработку сервером. Пример кода, добавленный в проект для настройки поиска статических файлов, представлен в листинге 2.2.

Листинг 2.2 – Параметр `STATICFILES_DIRS`

```
STATICFILES_DIRS = [  
    os.path.join(BASE_DIR, 'static')  
]
```

Установление соединения с базой данных осуществляется посредством конфигурации параметров переменной `DATABASES`, в которой задаются ключи, соответствующие используемой системе управления базами данных (СУБД). Конфигурационный массив включает следующие обязательные параметры:

- `ENGINE` – указывает драйвер или движок, используемый для взаимодействия с конкретной системой управления базами данных;

- NAME – задает наименование базы данных, к которой осуществляется подключение;
- USER – указывает имя пользователя, обладающего необходимыми правами для доступа к базе данных;
- PASSWORD – содержит пароль пользователя для аутентификации;
- HOST – определяет сетевой адрес сервера базы данных;
- PORT – задает номер порта, по которому осуществляется подключение к серверу базы данных.

Пример конфигурации подключения к базе данных представлен в листинге 2.3, демонстрирующем правильную структуру и синтаксис задания параметров для успешного установления соединения.

Листинг 2.3 – Пример подключения к базе данных

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'HOST': 's1-v-dbdev-1.main.eip',
        'NAME': 'cscd',
        'SCHEME': 'auth',
        'PORT': '5432',
        'USER': 'auth_user',
        'PASSWORD': 'auth_user',
    },
}
```

При реализации подключения нескольких баз данных в рамках проекта на Django необходимо настроить механизм маршрутизации (роутинг), обеспечивающий распределение запросов между указанными базами данных. Django предоставляет встроенные средства для определения маршрутов работы с различными базами данных, что позволяет управлять использованием конкретных баз данных для моделей и операций с ними. Данный механизм роутинга баз данных реализуется посредством определения специального класса маршрутизатора (router), который содержит методы, отвечающие за выбор базы данных для выполнения операций чтения, записи, а также для определения, к какой базе данных следует обращаться при

выполнении миграций. Конфигурация такого роутера позволяет гибко управлять распределением нагрузки и обеспечивать изоляцию данных между различными источниками. Пример настройки и определения правил роутинга представлен в листинге 2.4.

Листинг 2.4 – Роутинг баз данных

```
class DbRouter:

    def db_for_read(self, model, **hints):
        return model.database if hasattr(model, 'database') else
'default'

    def db_for_write(self, model, **hints):
        return model.database if hasattr(model, 'database') else
'default'

    def allow_relation(self, obj1, obj2, **hints):
        return True

    def allow_migrate(self, db, app_label, model_name=None, **hints):
        return True
```

Для взаимодействия с базами данных в рамках проекта на Django были разработаны классы моделей, которые служат отображением структуры таблиц базы данных в виде объектов Python. Модель представляет собой определение схемы данных, включающее описание полей, их типов и связей между различными моделями, что обеспечивает удобное управление данными и их целостностью.

Основные компоненты модели в Django включают:

- поля (Fields): определяют тип данных, который может храниться в базе данных для каждого конкретного атрибута модели. Например, CharField используется для хранения строковых значений, IntegerField — для целых чисел и так далее. Поля также могут содержать дополнительные параметры, такие как длина строки, обязательность заполнения и значения по умолчанию.
- связи (Relationships): Django поддерживает установление связей между моделями посредством различных типов полей: ForeignKey — для связи

"один ко многим", OneToOneField — для связи "один к одному", ManyToManyField — для связи "многие ко многим". Эти связи позволяют моделировать отношения между записями в различных таблицах базы данных.

- методы и свойства класса: Модели могут содержать собственные методы и свойства, предназначенные для выполнения операций с данными, таких как создание, получение, обновление или удаление объектов. Это обеспечивает инкапсуляцию бизнес-логики внутри модели.

Django предоставляет автоматизированные средства для генерации моделей на основе существующей схемы базы данных. В частности, команда `python manage.py inspectdb` позволяет автоматически создать классы моделей, анализируя структуру таблиц в базе данных и формируя соответствующие определения. При использовании этой команды без указания ключа `--database`, Django обращается к базе данных по умолчанию (ключ `'default'`), что упрощает интеграцию с существующими системами. Пример реализации класса модели приведен в листинге 2.5, демонстрирующем структуру и синтаксис определения моделей в Django [17].

Листинг 2.5 – Реализация класса модели

```
class FileCard(models.Model):
    database='fs'
    file_id = models.BigIntegerField(db_column='файл_ID',
primary_key=True, unique=False)
    archive_struct = models.CharField(max_length=255,
db_column='структура_архива')
    file_name = models.CharField(max_length=255,
db_column='имя_файла')
    file_size = models.CharField(max_length=50,
db_column='суммарный_размер')

    class Meta:
        managed = False
        db_table = '_файл_карточка'
```

Функции вызова процедур из базы данных добавляются в файл `views.py`, код которого представлен в приложении В. В данном файле определяются логические блоки, отвечающие за взаимодействие с базой данных, обработку

входных данных и формирование ответов для клиента. На листинге 2.6 показан пример вызова процедуры для получения сообщений по конкретному извещению [18].

Листинг 2.6 – Пример вызова процедуры для получения сообщений по конкретному извещению

```
def getMsgs(request):
    id_sed = request.GET.get('id_sed')
    json_msgs = []
    msgs_sql = '''
        DECLARE @RC int
        DECLARE @документ_id bigint
        SET @документ_id = %s
        EXECUTE @RC = dbo._mc49p_список_переписка @документ_id'''
    result = MC49PListMsgs.objects.raw(msgs_sql, [id_sed])
    for i in list(result):
        json_msgs.append({
            "author": i.author,
            "msg_data": i.msg_data,
            "msg_text": i.msg_text,
            "msg": i.msg,
            "msg_id": i.msg_id
        })
    return JsonResponse(json_msgs, safe=False)
```

Выделение ресурсов на этапе проектирования является ключевым аспектом разработки, обеспечивающим создание масштабируемых и адаптивных веб-приложений. Основные ресурсы, выделенные для программного модуля:

- close/ – переход на страницу просмотра договоров, направленных на закрытие;
- doc-card/<str:heading_name>/ [name='doc-card'] – просмотр карточки документа;
- api/dlist/ – получение договоров, направленных на удаление;
- api/ddlist/ – получение договоров из реестра;
- api/cma/ – получение договоров, направленных на закрытие;
- api/show-files/ – просмотр закрепленных за договором файлов;

- `api/send-doc-id-sed/` – получение списка документов для закрытия договоров;
- `api/get-msgs/` – просмотр переписки, закрепленной за извещением;
- `api/del-select-links/` – удаление связей у выбранных договоров;
- `api/checked-add/` – добавление выбранных договоров в реестр;
- `api/checked-del/` – удаление выбранных договоров из реестра;
- `api/del-table-filter/` – удаление файлов из реестра;
- `api/add-table-filter/` – добавление договоров в реестр;
- `api/del-link/` – удаление связей во всей таблице;
- `api/no-contract-list/` – получение списка номеров документов;

Django REST Framework (DRF) является мощным и гибким инструментом для разработки RESTful API на базе фреймворка Django. Он предоставляет широкий набор средств для реализации взаимодействия между клиентами и сервером, среди которых особое значение имеют сериализаторы и представления.

Сериализаторы в DRF выступают в качестве ключевых компонентов, обеспечивающих преобразование объектов модели или произвольных данных Python в форматы, пригодные для передачи по сети, такие как JSON или XML [4]. Они позволяют выполнять двунаправленное преобразование: из объектов Python в сериализованный формат и обратно. Помимо этого, сериализаторы предоставляют механизмы валидации входных данных, что обеспечивает целостность и корректность передаваемой информации. В рамках реализации бизнес-логики сериализаторы также управляют отображением данных — определяют, какие поля и в каком виде будут представлены клиенту. В приложении Г приведен код используемых сериализаторов.

Представления в DRF отвечают за обработку входящих HTTP-запросов и формирование ответов. Они управляют логикой взаимодействия с сериализаторами, определяя порядок выполнения операций — создание, чтение, обновление или удаление данных. Представления могут быть

реализованы как в функциональном стиле (функции), так и в объектно-ориентированном стиле (классы), что позволяет выбрать наиболее подходящий подход для конкретных задач.

2.3 Проектирование клиентской части

Разработка клиентской части приложения является важнейшим этапом создания современного программного продукта, обеспечивающим удобство и интуитивную понятность взаимодействия пользователя с системой [10]. В рамках данной стадии были выполнены следующие шаги:

- проектирование пользовательского интерфейса (UI): включает в себя разработку визуальных стилей, оформление элементов интерфейса, а также создание макетов и прототипов веб-страниц [7]. Целью является обеспечение эстетически привлекательного и одновременно функционального внешнего вида приложения, способствующего положительному пользовательскому опыту [15].
- оптимизация пользовательского опыта (UX): предполагает настройку поведения интерфейса для обеспечения корректной и предсказуемой работы системы. В этот процесс входит реализация механизмов проверки вводимых данных, обработка ошибок, создание удобных навигационных элементов и обеспечение высокой отзывчивости системы, что способствует повышению удовлетворенности пользователя [5].
- настройка взаимодействия между клиентской и серверной частями: включает разработку механизмов обмена данными посредством API или других протоколов связи, а также оптимизацию логики работы приложения для обеспечения эффективной передачи информации. Это обеспечивает надежное и своевременное взаимодействие между фронтендом и бэкендом, что критически важно для функционирования системы в целом.

Внешний вид главной страницы, содержащей список договоров, направленных на удаление, а также реестра, представлен на рисунке 2.3. В

верхней части страницы содержится меню перехода в другой раздел. Переходы между страницами, а также ожидание вывода данных оснащены появляющимися индикаторами загрузки, в случаях, когда получение актуальных данных занимает больше времени, чем требуется.

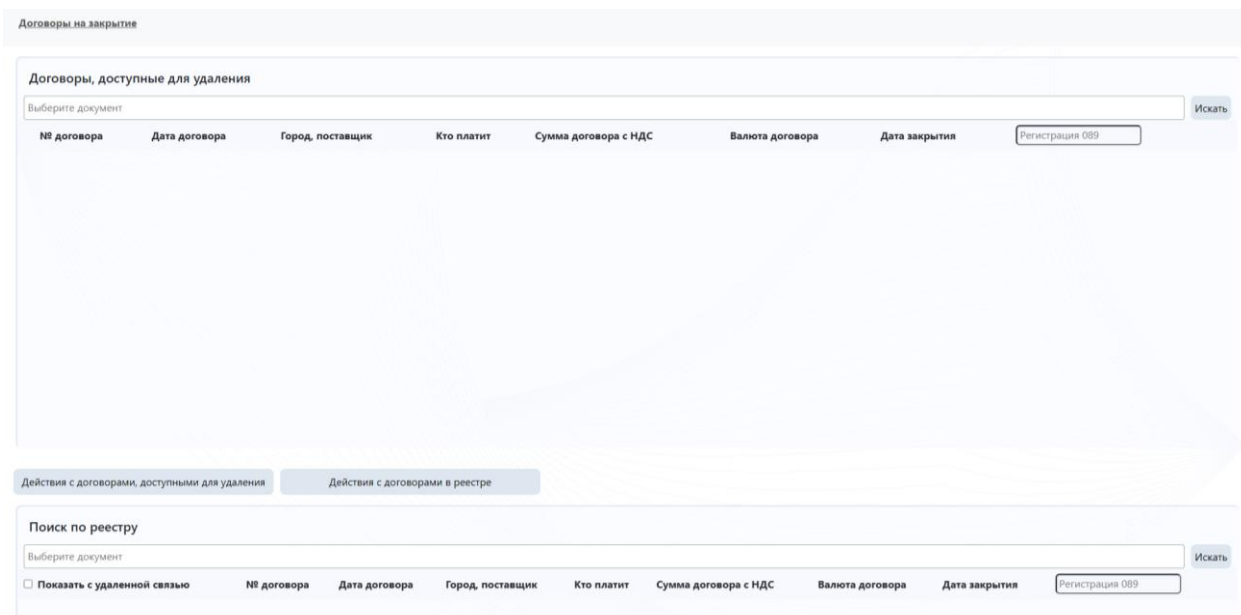


Рисунок 2.3 – Внешний вид главной страницы

На рисунке 2.4 представлен индикатор загрузки.

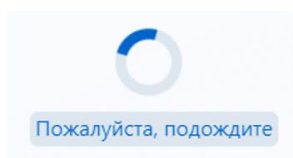


Рисунок 2.4 – Внешний индикатора загрузки

На рисунке 2.5 показана страница сайта, содержащая список договоров, направленных на закрытие.

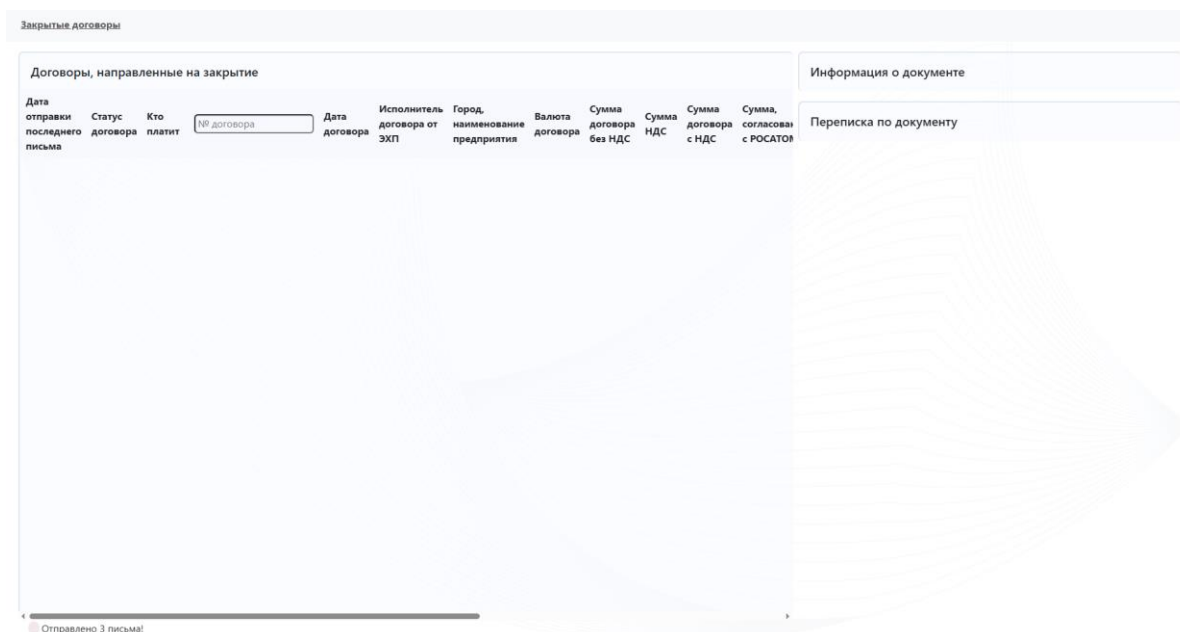


Рисунок 2.5 – Внешний вид страницы с договорами, направленными на закрытие

Для таблиц реализована механика постраничной навигации (пагинации), которая позволяет динамически определять и отображать количество страниц в зависимости от общего объема данных. На рисунке 2.6 представлена визуализация данного механизма. При использовании пагинации на каждой странице отображается фиксированное количество записей – 20 элементов. Такой подход предотвращает перегрузку интерфейса большим объемом данных, обеспечивая удобство взаимодействия пользователя с таблицей. Код реализации пагинации приведен в Приложении Д.

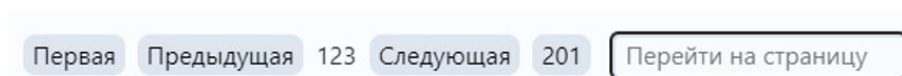


Рисунок 2.6 – Пагинация таблицы

Взаимодействие между клиентской и серверной частями системы реализуется посредством отправки формы с использованием метода GET. При наличии параметров поиска, таких как номер договора или регистрационный номер, серверное представление осуществляет фильтрацию данных и возвращает результат, соответствующий заданным критериям. Дополнительно предусмотрена возможность выбора порядка сортировки отображаемых

данных в таблице. Эта функциональность реализована через интерфейс, элементы которого представлены на рисунке 2.7.

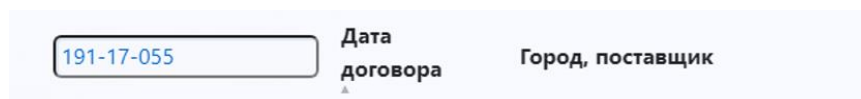


Рисунок 2.7 – Поиск и сортировка

Для каждого договора предусмотрена возможность просмотра детальной информации в виде карточки документа, внешний вид которой представлен на рисунке 2.8.

Просмотр договора с контрагентом



Рисунок 2.8 – Внешний вид карточки документа

На рисунке 2.9 представлена навигационная панель страницы карточки документа. Код реализации навигационной панели изложен в Приложении Е. Он включает структуру HTML-разметки и стилизацию при помощи CSS.



Рисунок 2.9 – Внешний вид навигационной панели

Как итог, в соответствии с техническим заданием на разработку программного модуля, в данном разделе были выполнены следующие задачи: спроектирована база данных, а также разработаны серверная и клиентская части приложения.

РАЗДЕЛ 3 ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЯ

3.1 Руководство пользователя

Данный программный продукт представляет собой веб-ориентированное приложение. Эта система предназначена для функционирования в браузере на рабочем месте пользователя, что обеспечивает удобство доступа и эксплуатации без необходимости установки дополнительного программного обеспечения на локальные устройства. В рамках архитектуры системы реализована модель клиент-сервер, где серверная часть обрабатывает запросы и управляет данными, а клиентская часть обеспечивает пользовательский интерфейс и взаимодействие с системой через стандартный веб-браузер. Минимальные технические требования к аппаратному обеспечению для успешной установки и функционирования данного веб-сервиса включают:

Процессор: рекомендуется использовать процессор типа Pentium II или более современное решение, способное обеспечить достаточную вычислительную мощность для обработки запросов и работы веб-сервиса без задержек. В случае более современных процессоров рекомендуется учитывать их совместимость с операционной системой и программным обеспечением.

Оперативная память: минимальный объем оперативной памяти составляет 4 Гб. Такой объем обеспечивает достаточную оперативность обработки данных и выполнение фоновых процессов без снижения производительности системы.

Жесткий диск: необходимое минимальное свободное пространство составляет 1 Гб.

Операционная система: допускается использование любой операционной системы, совместимой с серверным программным обеспечением и обеспечивающей стабильную работу веб-сервиса.

Запуск модуля осуществляется посредством перехода по специально подготовленной ссылке, которая размещается на рабочем месте пользователя

администратором системы или автоматически предоставляется через внутренние средства организации. Данная ссылка обеспечивает доступ к системе без необходимости локальной установки программных компонентов, что способствует быстрому внедрению и повышению удобства эксплуатации системы в условиях закрытой локально-вычислительной сети организации.

Главная страница данного веб-приложения представляет собой интерфейс, предназначенный для отображения списка закрытых договоров, срок хранения которых истек. При переходе на соответствующую HTML-страницу автоматически загружается таблица, содержащая перечень данных договорных документов. Визуальное оформление и структура страницы представлены на рисунке 3.1.

Договоры, доступные для удаления (1027)

Выберите документ

№ договора	Дата договора	Город, поставщик	Кто платит	Сумма договора с НДС	Валюта договора	Дата закрытия	Регистрация 089	Прикрепленные файлы
<input type="checkbox"/> 191-17-055-48/362	25.07.2019	село Косулино, ООО "АвтоСила"	Мы	70 400 000,00	рубли	30.12.2019	19-1-055-0098	Прикрепленные файлы
<input type="checkbox"/> 191-17-055-83/538	05.11.2019	Екатеринбург, ООО Торговая Компания "Райдо-Сервис"	Мы	1 480 892,48	рубли	30.12.2019	19-1-055-0211	Прикрепленные файлы
<input type="checkbox"/> 191-17-055-83/500	28.10.2019	Екатеринбург, ООО "Проекты и технологии"	Мы	2 765 999,52	рубли	30.12.2019	19-1-055-0200	Прикрепленные файлы
<input type="checkbox"/> 077-04/52	06.08.2019	Озёрок, ООО "СилКаМ"	Мы	1 345 632,40	рубли	30.12.2019	19-0-077-0074	Прикрепленные файлы
<input type="checkbox"/> 077-04/51	05.08.2019	Калуга, ООО "ЭРГА"	Мы	1 602 720,00	рубли	30.12.2019	19-0-077-0072	Прикрепленные файлы
<input type="checkbox"/> 2617-232-К	29.06.2017	Москва, Банк ГПБ (АО)	Мы	1 179 152 000,00	рубли	27.12.2019	17-0-089-0015	Прикрепленные файлы
<input type="checkbox"/> 191-17-055-32/795	01.10.2018	Екатеринбург, ООО "НПО "РЕГИТЕ"	Мы	21 872 954,40	рубли	27.12.2019	19-0-055-0317	Прикрепленные файлы

Действия с договорами, доступными для удаления Действия с договорами в реестре

Поиск по реестру

Выберите документ

☐ Показать с удаленной связью № договора Дата договора Город, поставщик Кто платит Сумма договора с НДС Валюта договора Дата закрытия Регистрация 089 Искать

Рисунок 3.1 – Главная страница сайта

В заголовке таблицы в круглых скобках указано общее количество договоров, содержащихся в текущем списке. В данном конкретном случае это число равно одной тысяче двадцати семи строкам, что позволяет пользователю быстро ориентироваться в объеме доступных данных и оценивать масштаб работы с документами.

Для удобства анализа реализована возможность сортировки таблицы по следующим ключевым полям:

- «Дата договора»;

- «Дата закрытия»;
- «Сумма договора с НДС».

Кроме того, реализована функция поиска для быстрой фильтрации записей по следующим полям:

- «Номер договора»;
- «Регистрация в 089 отделе».

Поиск договоров по их номеру осуществляется при помощи выпадающего списка с возможностью выбора нескольких номеров. Пример использования строки поиска приведен на рисунке 3.2.

Рисунок 3.2 – Строка поиска

Пример реализации фильтрации данных по полю «Номер договора» и последующей сортировки результатов по убыванию значения поля «Сумма договора с НДС» проиллюстрирован на рисунке 3.3. В результате применения указанных фильтров и сортировки общее число отображаемых договоров сократилось до пяти записей.

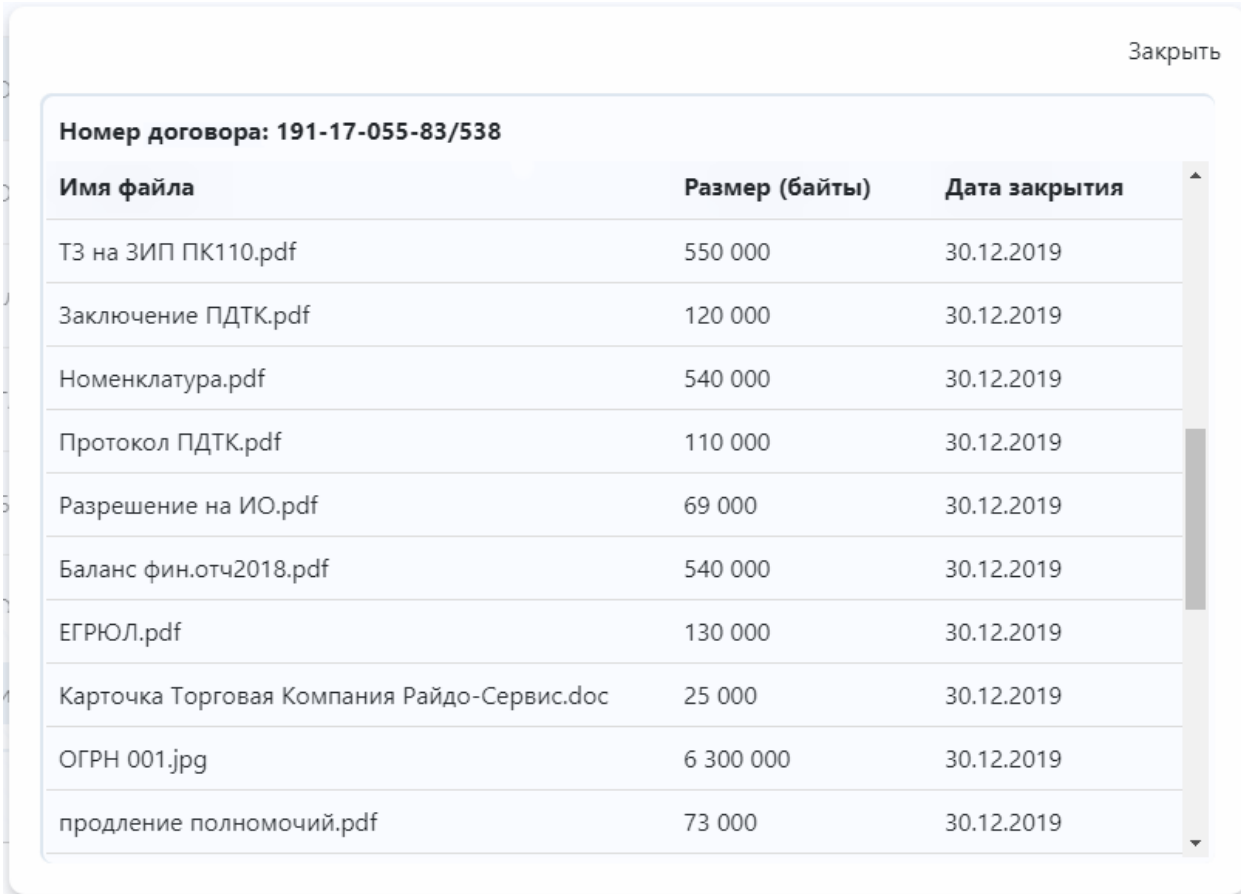
Договоры, доступные для удаления (5)

№ договора	Дата договора	Город, поставщик	Кто платит	Сумма договора с НДС	Валюта договора	Дата закрытия	Регистрация 089	
<input type="checkbox"/> 191-17-055-40/126	28.03.2019	Челябинск, ООО "Спецпроминструмент"	Мы	1 991 907,38	рубли	13.08.2019	19-2-055-0029	Прикрепленные файлы
<input type="checkbox"/> 191-17-055-40/73	25.03.2019	Екатеринбург, ООО "Сперанца"	Мы	949 749,72	рубли	11.09.2019	19-2-055-0022	Прикрепленные файлы
<input type="checkbox"/> 191-17-055-17/132	04.04.2019	Екатеринбург, ООО "ПФГ "АШС"	Мы	878 160,04	рубли	03.07.2019	19-0-055-0087	Прикрепленные файлы
<input type="checkbox"/> 191-03-014-09/186	28.03.2019	Лесной, ИП Шостер Наиля Хатимовна	Мы	299 700,00	рубли	13.05.2019	19-0-014-0017	Прикрепленные файлы
<input type="checkbox"/> 191-19-031-20/6	28.03.2019	Лесной, МБУДО ДМШ	Мы	246 000,00	рубли	28.08.2019	19-0-031-0008	Прикрепленные файлы

Рисунок 3.3 – Фильтрация данных в таблице

Дополнительно предусмотрена возможность просмотра прикрепленных файлов, связанных с каждым договором. При нажатии на кнопку «Прикрепленные файлы» открывается всплывающее диалоговое окно,

содержащее список всех файлов, закрепленных за выбранным договором. Визуальное оформление этого окна и его содержимое представлены на рисунке 3.4.



Номер договора: 191-17-055-83/538		
Имя файла	Размер (байты)	Дата закрытия
ТЗ на ЗИП ПК110.pdf	550 000	30.12.2019
Заключение ПДТК.pdf	120 000	30.12.2019
Номенклатура.pdf	540 000	30.12.2019
Протокол ПДТК.pdf	110 000	30.12.2019
Разрешение на ИО.pdf	69 000	30.12.2019
Баланс фин.отч2018.pdf	540 000	30.12.2019
ЕГРЮЛ.pdf	130 000	30.12.2019
Карточка Торговая Компания Райдо-Сервис.doc	25 000	30.12.2019
ОГРН 001.jpg	6 300 000	30.12.2019
продление полномочий.pdf	73 000	30.12.2019

Рисунок 3.4 – Модальное окно со списком файлов

Для повышения удобства работы с перечнем договоров каждый элемент строки сопровождается чекбоксом. Пример выбранных договоров проиллюстрирован на рисунке 3.5. Эти чекбоксы позволяют пользователю выбрать один или несколько конкретных договоров для последующей их отправки в реестр.

	№ договора	Дата договора	Город, поставщик	Кто платит
<input checked="" type="checkbox"/>	191-17-055-40/126	28.03.2019	Челябинск, ООО "Спецпроминструмент"	Мы
<input checked="" type="checkbox"/>	191-17-055-40/73	25.03.2019	Екатеринбург, ООО "Сперанца"	Мы
<input type="checkbox"/>	191-17-055-17/132	04.04.2019	Екатеринбург, ООО "ПФГ "АШС"	Мы
<input type="checkbox"/>	191-03-014-09/186	28.03.2019	Лесной, ИП Шостер Наиля Хатимовна	Мы
<input checked="" type="checkbox"/>	191-19-031-20/6	28.03.2019	Лесной, МБУДО ДМШ	Мы

Рисунок 3.5 – Выбранные договоры

Чтобы загрузить договоры в реестр, пользователю необходимо нажать на кнопку «Действия с договорами, доступными для удаления», внешний вид которой, с выпадающим списком, представлен на рисунке 3.6. В выпадающем списке предложены следующие опции:

- «Загрузить в реестр (выбранное)» – выгружает в реестр только выбранные договоры. В случае, если ничего не выбрано, выведется ошибка, как показано на рисунке 3.7;
- «Загрузить в реестр (все в таблице)» – выгружает в реестр все текущее содержимое таблицы, предварительно запрашивая у пользователя подтверждение, проиллюстрированное на рисунке 3.8.

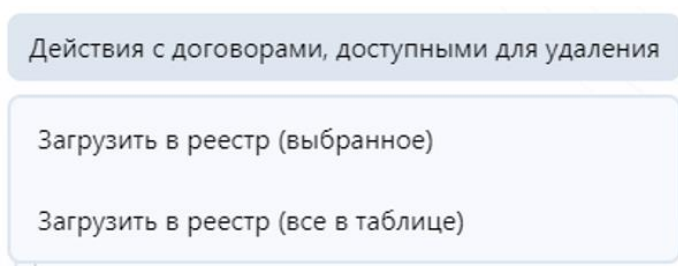


Рисунок 3.6 – Действия с договорами, доступными для удаления

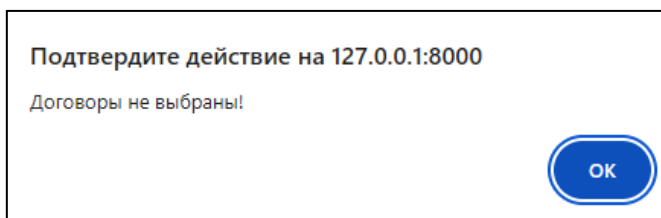


Рисунок 3.7 – Вывод ошибки пользователю

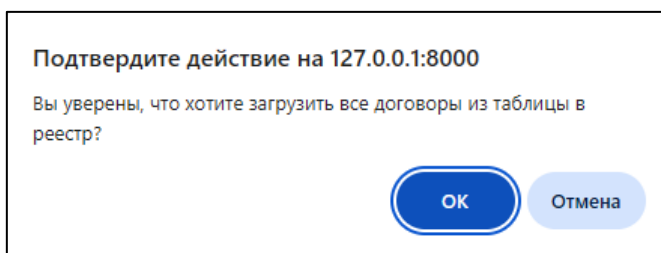


Рисунок 3.8 – Вывод подтверждения пользователю

Договоры, недавно отправленные в реестр, могут быть легко обнаружены благодаря появляющейся кнопке для просмотра списка их номеров. На рисунке 3.9 представлен пример такого списка.

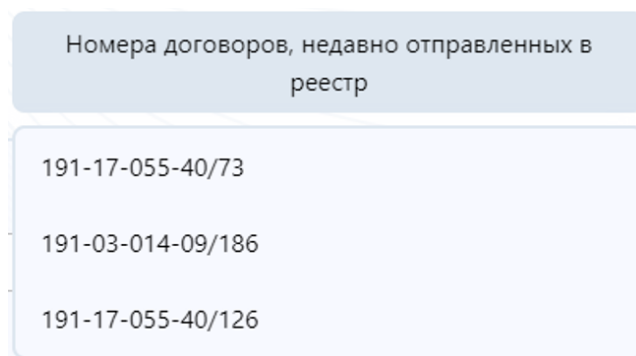


Рисунок 3.9 – Номера договоров, недавно отправленных в реестр

Для удобного взаимодействия с реестром также реализованы две строки поиска и фильтрация по числовым полям. На рисунке 3.10 показан пример поиска недавно загруженного в реестр договора по его номеру.

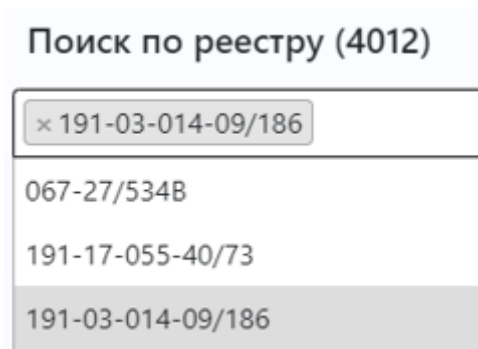


Рисунок 3.10 – Поиск по реестру

Список действий с реестром можно увидеть, нажав кнопку «Действия с договорами в реестре», внешний вид которой, с выпадающим списком, представлен на рисунке 3.11. В выпадающем списке предложены следующие опции:

- «Показать договоры без удаленной связи» – для быстрой фильтрации и отображения только тех договорных документов, которые можно удалить из реестра.
- «Убрать из реестра (выбранное)» – выбранные записи будут исключены из реестра и возвращены обратно в исходную таблицу закрытых договоров. В случае, если ничего не выбрано, выведется ошибка;
- «Убрать из реестра (все в таблице)» – выгружает из реестра все текущее содержимое таблицы, предварительно запрашивая у пользователя подтверждение;

- «Удалить связь (выбранное)» – запускает процедуру удаления связи над wybranными договорами. В случае, если ничего не выбрано, выведется ошибка;
- «Удалить связь (все)» – запускает процедуру удаления связи со всеми договорами в реестре, над которыми еще не была применена ранее, предварительно запрашивая у пользователя подтверждение на выполнение.

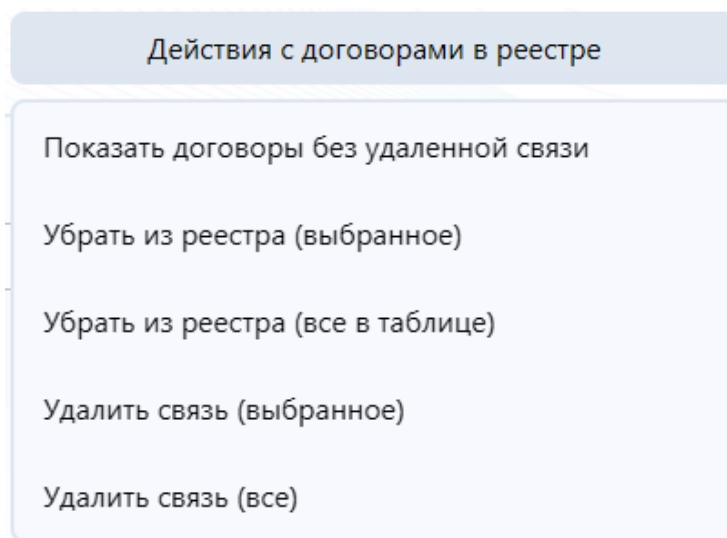


Рисунок 3.11 – Действия с договорами в реестре

Для каждого договора, у которого еще не была удалена связь в базе данных, присутствует активный чекбокс. Выбранные в реестре договоры представлены на рисунке 3.12. Эти чекбоксы позволяют выбрать один или несколько договоров для последующих операций.

Выберите документ				
<input type="checkbox"/> Показать с удаленной связью	№ договора	Дата договора	Город, поставщик	Кто платит
<input checked="" type="checkbox"/>	191-17-055-39/44 1	04.10.2019	Екатеринбург, ООО "ПромКомплектЦентр"	Мы
<input checked="" type="checkbox"/>	191-03-014-09/17	28.01.2019	Новоуральск, ООО "ОРБ"	Нам
<input type="checkbox"/>	191-17-055-40/73	25.03.2019	Екатеринбург, ООО "Сперанца"	Мы
<input type="checkbox"/>	191-17-055-40/12 6	28.03.2019	Челябинск, ООО "Спецпроминструмент"	Мы

Рисунок 3.12 – Выбранные в реестре договора

Для отображения договоров с удаленной связью, которые уже нельзя убрать из реестра, предусмотрена возможность фильтрации с помощью

специального флажка. Этот флажок-фильтр, как показано на рисунке 3.13, позволяет пользователю выбрать отображение только тех договоров, у которых связь в базе данных была удалена. Если же пользователь снимает данный флажок, система отображает данные как показано на рисунке 3.12. Это обеспечивает гибкость в управлении и просмотре данных, позволяя быстро переключаться между разными группами договоров.

Выберите документ				
<input checked="" type="checkbox"/> Показать с удаленной связью	№ договора	Дата договора	Город, поставщик	Кто платит
<input type="checkbox"/>	055-57/609	14.05.2018	Екатеринбург, ООО "СТА-трейд"	Нам
<input type="checkbox"/>	191-17-055-35/611	28.05.2018	Москва, ООО "СтройИнженерМонтаж"	Мы

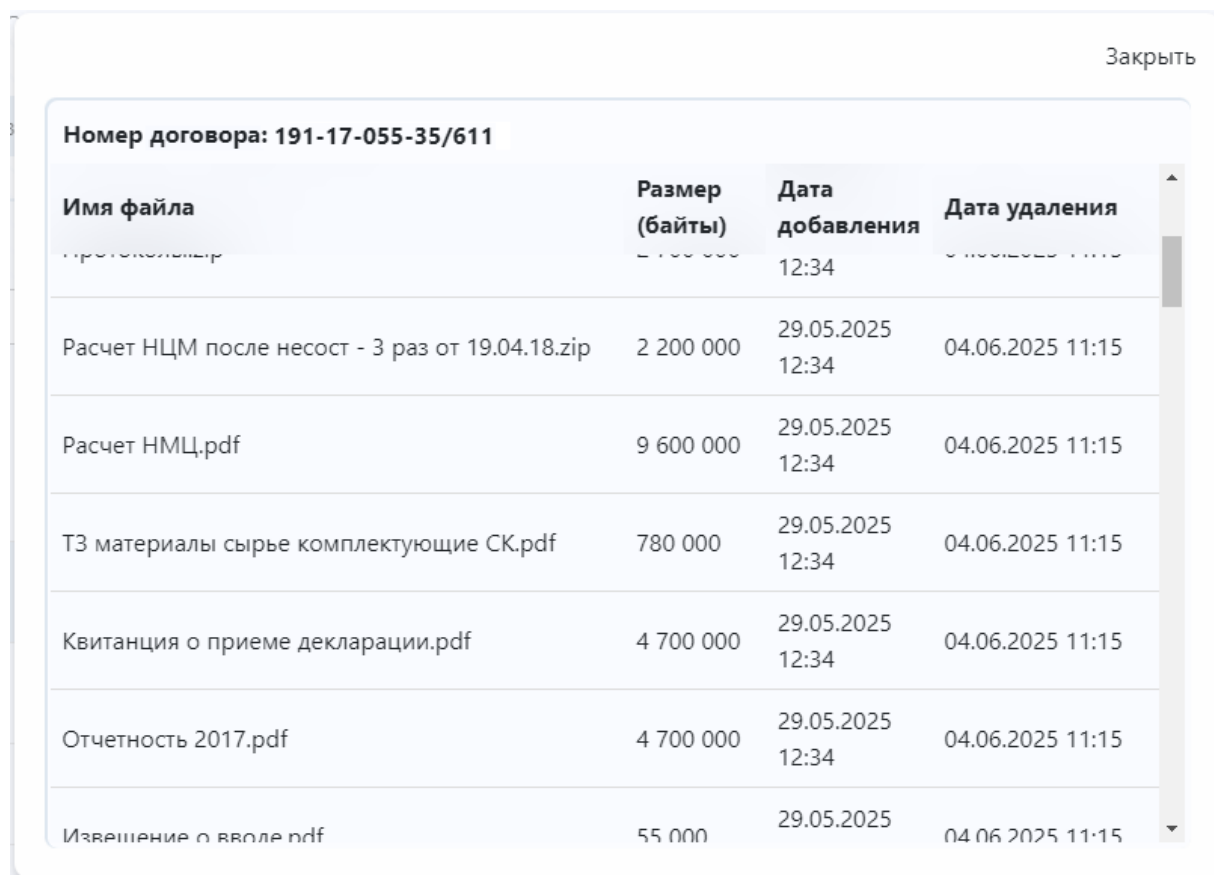
Рисунок 3.13 – Договоры с удаленной связью

Дополнительно, для договоров с неудаленной связью, в модальном окне со списком закрепленных за договором файлов столбец «Дата удаления» будет пустым, что подтверждается изображением на рисунке 3.14.

Номер договора: 191-17-055-85/538				Заккрыть
Имя файла	Размер (байты)	Дата добавления	Дата удаления	
Запрос обеспечения договора.pdf	180 000	06.03.2025 11:23		
Запрос ТКП от 18.09.2019 №191-055-06485.pdf	1 100 000	06.03.2025 11:23		
Извещение о проведении МЗ.pdf	730 000	06.03.2025 11:23		
Коммерческое предложение Медиа Плюс.pdf	400 000	06.03.2025 11:23		
Коммерческое предложение Райдо-сервис.pdf	500 000	06.03.2025 11:23		
Коммерческое предложение Урал-бюро.р	730 000	06.03.2025		

Рисунок 3.14 – Модальное окно реестра с договорами без удаленной связи

В противоположность этому, у договоров с удаленной связью данный столбец заполнен датой удаления связи, что видно на рисунке 3.15.



Имя файла	Размер (байты)	Дата добавления	Дата удаления
Протокол...	1 700 000	29.05.2025 12:34	04.06.2025 11:15
Расчет НЦМ после несост - 3 раз от 19.04.18.zip	2 200 000	29.05.2025 12:34	04.06.2025 11:15
Расчет НМЦ.pdf	9 600 000	29.05.2025 12:34	04.06.2025 11:15
ТЗ материалы сырье комплектующие СК.pdf	780 000	29.05.2025 12:34	04.06.2025 11:15
Квитанция о приеме декларации.pdf	4 700 000	29.05.2025 12:34	04.06.2025 11:15
Отчетность 2017.pdf	4 700 000	29.05.2025 12:34	04.06.2025 11:15
Извещение о вводе.pdf	55 000	29.05.2025	04.06.2025 11:15

Рисунок 3.15 – Модальное окно реестра с договорами с удаленной связью

Договоры, недавно удаленные из реестра, также могут быть легко обнаружены благодаря появляющейся кнопке для просмотра списка их номеров, как показано на рисунке 3.16.

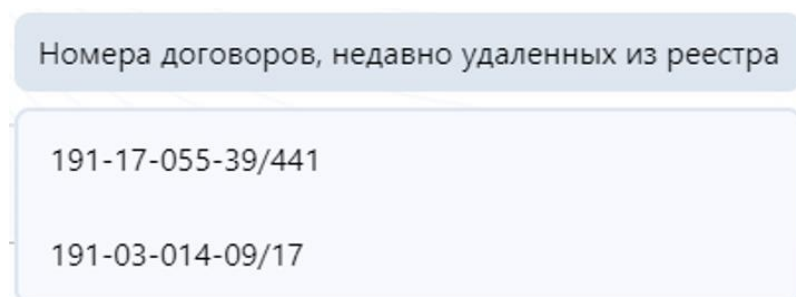


Рисунок 3.16 – Номера договоров, недавно удаленных из реестра

При переходе на вкладку «Договоры на закрытие» пользователь получает доступ к таблице, содержащей список договоров, направленных на процедуру электронного закрытия, что изображено на рисунке 3.17.

Визуальное выделение договоров красным цветом указывает на то, что по ним уже было отправлено три извещения. Такой цветовой маркер помогает быстро идентифицировать договоры, требующие внимания или дополнительных действий.

Закрытые договора

Документы, направленные на закрытие (149)

Дата отправки последнего письма	Статус договора	Кто платит	№ договора	Дата договора	Исполнитель договора от ЭХП	Город, наименование предприятия	Валюта договора	Сумма договора без НДС	Сумма НДС	Сумма договор НДС
14.06.2024	ДОГОВОР ИСПОЛНЕН 100 %	Мы	4-3527/22	06.09.2022	Шарина Екатерина Юрьевна	Екатеринбург, АО "Уралсвергаз-независимая газовая компания" (ИНН 6604008860; КПП 667801001)	рубли	20 769 469,62	4 153 893,92	24 923 363,54
14.06.2024		Мы	1322 GP 0077	01.08.2022	Черепанова Валерия Евгеньевна	Москва, АО "Согаз" (ИНН 7736035485; КПП 770801001)	рубли	70 644,67	0,00	70 644,67
14.06.2024	ДОГОВОР ИСПОЛНЕН	Нам	191-17-013-12/02	26.01.2022	Черепанова Валерия Евгеньевна	Лесной, МКУ "ФКУ" (ИНН 6630012782; КПП 663001001)	рубли	25 545,00	5 109,00	30 654,00
14.06.2024	ДОГОВОР ИСПОЛНЕН	Нам	191-17-013-12/08	17.01.2022	Черепанова Валерия Евгеньевна	Москва, АО "ФЦНИВТ "СНПО "Элерон" (ИНН 7724313681; КПП 772401001)	рубли	83 330,00	16 666,00	99 996,00

Отправлено 3 письма!

Информация о документе

№ Документа: ЭЛ-Зак.Дог.-24-2390

Дата документа: 14.06.2024

Текст документа (письма):

1

Уважаемые руководители/кураторы/исполнители! По данным задачи МС49Р "Автоматизированная система управления договорной деятельностью" по договору **№84-3527/22**, дата 06.09.2022, заключенному с контрагентом (918 АО "Уралсвергаз-независимая газовая компания" Екатеринбург), ответственное подразделение 006, ФИО куратора/исполнителя Шарина Екатерина Юрьевна отсутствует задолженность на текущую дату и завершён период расчетов по договору. Прошу проверить обстоятельства, необходимые для закрытия договора в соответствии с пунктами 9.3.3-9.3.5 СТО ЖО8.0052-2020, и своевременно направить договор на закрытие в задаче МС49Р. Напоминаю, что не закрытие исполненного (распоряженного) договора является основанием для привлечения исполнителя по договору к мерам дисциплинарной ответственности (пункты 1, 16 Перечня нарушений исполнительской дисциплины).

Переписка по документу

Рисунок 3.17 – Договоры, направленные на закрытие

При одинарном клике по любому договору в таблице на боковой панели автоматически отображается подробная информация о выбранном документе.

В этом разделе пользователь может увидеть:

- Детали самого договора;
- Количество отправленных извещений;
- Содержание каждого из них;
- Переписку, закрепленную за соответствующим извещением (если таковая имеется).

Это обеспечивает удобный и быстрый доступ ко всей связанной с договором информацией без необходимости перехода в отдельные разделы системы, как видно на рисунке 3.18.

Информация о документе

№ Документа: ЭЛ-ERP-авт. р-23-2203

Дата документа: 18.12.2023

Текст документа (письма):

123

Уважаемые руководители/кураторы/исполнители! По данным задачи МС49Р "Автоматизированная система управления договорной деятельностью" по договору №1322 GP 0077 , дата 01.08.2022, заключенному с контрагентом (23690 АО "Согаз" Москва) , ответственное подразделение 013, ФИО куратора/исполнителя Черепанова Валерия Евгеньевна отсутствует задолженность на текущую дату и завершен период расчетов по договору. Прошу проверить обстоятельства, необходимые для закрытия договора в соответствии с пунктом 9.3.3 СТО Ж08.0052-2020, и своевременно направить договор на закрытие в задаче МС49Р. Напоминаю, что не закрытие исполненного (расторгнутого) договора является основанием для привлечения исполнителя по договору к мерам дисциплинарной ответственности (пункты 1, 16 Перечня нарушений исполнительской дисциплины работником, приложение №2 к

Переписка по документу

(ERP) ERP

Уважаемые руководители и ответственные исполнители! Сообщение об инициировании закрытия данного договора направлено вам ПОВТОРНО! Прошу обратить внимание, что при отсутствии с вашей стороны каких-либо действий по закрытию договора, с февраля 2024 года контроль закрытия договоров, по которым было направлено два сообщения об инициировании закрытия (первичное и повторное), будет вестись в АСКИОП.

18.12.2023 14:40

Рисунок 3.18 – Информация по договору с перепиской

Строки в таблице также можно сортировать по следующим ключевым полям:

- Дата отправки последнего письма;
- Дата договора;
- Сумма договора с НДС;
- Сумма НДС;
- Сумма договора без НДС;
- Сумма, согласованная с Росатомом;

- Кол-во отправленных писем;
- Кол-во нарушений сроков поставки;
- Кол-во претензий по качеству.

Кроме того, реализована функция поиска для быстрой фильтрации записей по следующим полям:

- Номер договора;
- Код поставщика;
- Регистрация в 089 отделе.

Пример реализации фильтрации данных по полю «Номер договора» и последующей сортировки результатов по убыванию значения поля «Сумма договора без НДС». После выполнения фильтрации система автоматически обновила таблицу, показав только те договоры, которые соответствуют заданному критерию, что видно на рисунке 3.19.

<input type="text" value="191"/>	Дата договора	Исполнитель договора от ЭХП	Город, наименование предприятия	Валюта договора	Сумма договора без НДС
191-17-055-81/315	05.06.2023	Волкова Ирина Станиславовна	Москва, ООО "РУС" (ИНН 7731301060; КПП 773101001)	рубли	79 473 234,00
191-17-055-35/393	21.06.2022	Павлов Илья Сергеевич	Первоуральск, ООО Компания "Стройметалл" (ИНН 6684012468; КПП 668401001)	рубли	36 170 787,24

Рисунок 3.19 – Договоры, направленные на закрытие

При выполнении двойного клика мыши по договору в любом разделе системы пользователь получает возможность перейти к карточке документа, которая содержит детальную и структурированную информацию по выбранному договору. Эта карточка служит централизованным информационным порталом, предоставляющим всесторонний обзор ключевых аспектов договорных отношений, и обеспечивает удобство навигации и анализа данных.

В частности, в карточке отображаются следующие разделы:

- Основные реквизиты договора;
- Условия договора;
- Контакты;
- Обеспечение договора;
- Закупки;
- Условия расчетов/поставки.

В разделе «Основные реквизиты договора», как показано на рисунке 3.20, представлена информация о важнейших идентификационных и юридических параметрах документа, таких как тип документа, его уникальный номер, идентификатор государственного контракта или договора (соглашения), дата вступления договора в силу (которая может отличаться от даты его подписания), номер заявки и дата её подачи, а также сведения о наличии дополнительных соглашений. Кроме того, здесь указываются стороны договора: кто осуществляет оплату, дата начала действия договора, дата окончания срока его действия, а также сведения о продавце, грузоотправителе и получателе платежа.

Тип документа Договор

№ документа 191-17-055-38/901

Дата договорного документа 14.01.2022

Кто платит Мы Начало действия договора 14.01.2022 Окончание действия договора 29.08.2022

Продавец (№ по справочнику)	24566	ООО "МЕТМАШ" ИНН=7714371564 КПП=771401001 Юридический адрес: 125040, Россия, г.Москва, ул.Скаковая 32, строение 2, пом IX, Эт.8, К. 21-23 Почтовый адрес: 125040, Россия, Москва, ул.Скаковая 32, строение 2
Грузоотправитель (№ по справочнику)	24566	ООО "МЕТМАШ" ИНН=7714371564 КПП=771401001 Юридический адрес: 125040, Россия, г.Москва, ул.Скаковая 32, строение 2, пом IX, Эт.8, К. 21-23 Почтовый адрес: 125040, Россия, Москва, ул.Скаковая 32, строение 2
Получатель платежа (№ по справочнику)	24566	ИНН=7714371564 КПП=771401001 Юридический адрес: 125040, Россия, г.Москва, ул.Скаковая 32, строение 2, пом IX, Эт.8, К. 21-23 Почтовый адрес: 125040, Россия, Москва, ул.Скаковая 32, строение 2

Рисунок 3.20 – Основные реквизиты договора

Раздел «Условия договора» содержит подробную информацию о финансовых аспектах и предметной области договора, что представлено на рисунке 3.21. В частности, здесь представлены код финансирования, сумма сделки в рублях, вид договора (например, поставка, оказание услуг), предмет договора (описание товаров или услуг), момент фактической передачи продукции или товаров, дополнительные признаки или особенности сделки. Также указываются валютные параметры: валюта сделки и курс валюты на момент заключения договора (в рублях), суммы без НДС, суммы НДС и итоговая сумма с НДС. Важными элементами являются параметры включения суммы дополнительного соглашения в итоговую сумму договора, форма расчетов (предоплата, постоплата и т.д.), цель заключения договора (например, выполнение государственного задания или коммерческая сделка), статусы «Наш» и «Контрагента», условный номер контракта, номера доходных договоров и дополнительные реквизиты: номера договоров и даты их заключения.

Вид договора Поставка товаров

Предмет договора

Поставка поковок 5ХНМ, сталь 20

Момент фактической передачи продукции, товаров выборка (самовывоз)

Доп. признак П - Продукция, товары **Валюта** рубли

Курс валюты на момент заключения договора (в рублях) 1.0

Сумма без НДС 7 418 174,00 **Сумма НДС** 1 483 634,00 **Сумма с НДС** 8 901 809,00

Форма расчетов денежные средства

Цель заключения договора

Выпуск товарной продукции

Статус Наш Покупатель **Статус контрагента** Поставщик (продавец)

Тип документа	№ договора	Дата договора	№ доп. соглашения	Дата доп. соглашения	Дата начала	Дата окончания
Договор	191-17-055-38/901				14.01.2022	29.08.2022

Рисунок 3.21 – Условия договора

Раздел «Контакты» предназначен для отображения контактных данных ответственных лиц по договору с обеих сторон, его можно увидеть на рисунке 3.22. В нем представлены сведения о кураторе договора от контрагента или подписавшем документ лице: должность, ФИО, телефон и электронная почта. Аналогичные данные предоставляются для исполнителя от ЭХП — ФИО и контактные телефоны. Также указывается куратор со стороны ЭХП с соответствующими контактными данными. В разделе отображается информация о подразделениях организации или структурных единицах, для которых данный договор является доступным или актуальным.

Куратор договора от контрагента или лицо, подписавшее договор

Должность Менеджер

ФИО Гусак Александр

Телефон 84951034587

ФИО ГД gusak-91@mail.ru

Исполнитель договора от ЭХП

ФИО Павлов Илья Сергеевич

Телефон 95295

Куратор договора от ЭХП

ФИО Павлов Илья Сергеевич

Телефон 9-52-95

Подразделения, для которых доступен договор

055; 058;

Рисунок 3.22 – Контакты

Раздел «Обеспечение» содержит сведения о планируемых и фактических видах обеспечения исполнения обязательств по договору. В нем отображаются виды обеспечения с указанием их статуса — запланированные или фактические виды обеспечения исполнения обязательств, что видно на рисунке 3.23.

Планируемый вид обеспечения (из WC07P)						
Решение	Вид обеспечения	Сумма обеспечения, руб.		Срок действия обеспечения		
	Обеспечение возврата аванса	0,00		01.01.1900		

Фактический вид обеспечения						
Вид обеспечения	Форма обеспечения	Номер документа обеспечения	Дата документа обеспечения	Сумма обеспечения, руб.	Срок действия обеспечения	Лицо, предоставившее обеспечение
Обеспечение исполнения договора	Обеспечительный платеж	1429	24.12.2021	178 036,00	14.05.2022	Москва, ООО "МЕТМАШ"

Рисунок 3.23 – Обеспечение

Раздел «Закупки» предоставляет информацию о закупочных процедурах в рамках данного договора. Здесь указывается способ закупки (например, аукцион), номер позиции ГПЗ, пункт ЕОСЗ и основания для выбора конкретного контрагента, как видно на рисунке 3.24.

Способ закупки

Номер позиции в ГПЗ

Пункт ЕОСЗ

Основание для выбора контрагента

протокол №33321890ПО1-3

Рисунок 3.24 – Закупки

Раздел «Условия расчетов/поставки» содержит параметры финансовых расчетов и условий поставки товаров или услуг по договору. В частности, указываются процент авансирования и срок оплаты аванса в днях (с указанием конкретных дат). Также приводятся условия оплаты: сроки окончательного расчета (в днях), даты оплаты по условиям договора; сроки поставки товаров

или услуг — количество дней до даты поставки; условия оплаты при завершении поставки; а также параметры закрытия аванса и проценты окончательного расчета, что можно увидеть на рисунке 3.25.

Процент окончательного расчета	Срок оплаты (дни)	Дни	Срок оплаты (дата)	Условия оплаты
100,000	15	рабочие/банковские		3.6 подписание документов, подтверждающих исполнение обязательств по поставке товара (работы, услуги); товарная накладная (ТОРГ-12), акт приемки-сдачи выполненных работ, оказанных услуг

Срок поставки (дни)	Дни	Срок поставки (дата)	Условия оплаты
0	календарные	15.03.2022	4.7 не позднее указанной даты

Рисунок 3.25 – Условия расчетов/поставки

3.2 Планируемый эффект от внедрения

Разработка программного модуля для формирования реестра договоров, подлежащих удалению из баз данных после завершения сроков хранения, представляет собой важный этап модернизации системы управления договорной деятельностью. Этот модуль является ключевым компонентом стратегии перехода на импортонезависимое программное обеспечение и обеспечивает реализацию принципов кроссплатформенной совместимости системы. Внедрение данного модуля способствует повышению эффективности работы с данными, а также обеспечивает соответствие современным требованиям информационной безопасности и нормативным актам, регулирующим хранение и уничтожение документов.

Создание данного программного модуля обусловлено необходимостью автоматизации процессов учета и своевременного удаления устаревших договорных документов, что позволяет снизить нагрузку на информационные ресурсы, повысить производительность системы и обеспечить актуальность хранимых данных. В рамках реализации проекта предусмотрена возможность

интеграции модуля как в текущую инфраструктуру, так и в будущие версии системы, что обеспечивает гибкость и масштабируемость решений.

Внедрение разработанного программного модуля может быть осуществлено сразу после завершения его разработки и успешной защиты дипломного проекта. Такой подход позволит оперативно подготовить инфраструктуру к переходу на импортонезависимые операционные системы, такие как Astra Linux, которые широко используются в государственных структурах и организациях с повышенными требованиями к информационной безопасности. Реализация данного сценария обеспечит беспрепятственный переход на новые платформы без необходимости значительных доработок существующих системных компонентов. Альтернативным сценарием является ожидание завершения полной миграции всей системы управления договорной деятельностью на современную веб-платформу. После этого интеграция разработанного модуля станет частью комплексного обновления системы, что позволит обеспечить единый интерфейс для пользователей и повысить уровень автоматизации процессов. Основные ожидаемые эффекты от внедрения данного программного модуля включают следующие преимущества:

- Повышение доступности и удобства использования: благодаря реализации веб-интерфейса пользователи смогут работать с модулем независимо от операционной системы.
- Упрощение процессов администрирования: автоматизация формирования реестра договоров для удаления ускорит процессы подготовки документов к удалению и обеспечит соблюдение нормативных требований по срокам хранения.
- Создание задела для дальнейшей модернизации системы: разработка данного модуля закладывает основу для последующих обновлений и расширений функционала системы управления договорной деятельностью. В частности, это упростит миграцию других

компонентов информационной системы в веб-среду, что повысит ее гибкость, масштабируемость и устойчивость к внешним воздействиям.

- Обеспечение соответствия современным стандартам информационных технологий: переход на импортонезависимое программное обеспечение способствует снижению зависимости от иностранных поставщиков программных решений, повышает уровень информационной безопасности и обеспечивает выполнение требований государственных нормативных актов по защите информации.

В рамках выполнения технического задания на разработку программного модуля были поставлены конкретные задачи, направленные на обеспечение его функциональной полноты и удобства использования. В частности, было подготовлено руководство пользователя. Это позволяет обеспечить эффективное внедрение решения в рабочие процессы организации без необходимости дополнительной подготовки специалистов. Кроме того, в рамках проекта был проведен анализ ожидаемого эффекта от внедрения модуля.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы были изучены предметная область, основные термины, инструментальные средства, связанные с системой управления договорной деятельностью. Состоялось знакомство с ФГУП «Комбинат «Электрохимприбор» и частью его бизнес-процессов. В результате были созданы модели бизнес-процессов и сделан вывод о необходимости создания приложения, предназначенного для своевременного удаления документов с завершенным сроком хранения.

Была изучена база данных со всеми необходимыми сущностями и атрибутами, были реализованы процедуры и представления, необходимые для выполнения бизнес-процесса. На основе структурных элементов базы данных были созданы модели Django — серверной части информационной системы, что позволило реализовать все необходимые бизнес-функции. В рамках разработки пользовательского интерфейса были созданы шаблоны страниц, обеспечивающие удобное взаимодействие пользователя с системой. Для повышения удобства и эстетики интерфейса применялись современные веб-технологии: JavaScript для динамического взаимодействия, CSS для стилизации элементов и Bootstrap 5 — популярный фреймворк для быстрого создания адаптивных и привлекательных интерфейсов.

В результате выполнения всех поставленных задач было создано импортонезависимое прикладное веб-приложение, полностью отвечающее требованиям технического задания. Разработанное приложение удобно в использовании и расширяемо, что обеспечивает его эффективную интеграцию в существующую инфраструктуру организации и способствует автоматизации процессов управления сроками хранения договорных документов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бевзюк Е.А. Регламентация и нормирование труда: учебное пособие для бакалавров / Бевзюк Е.А., Попов С.В. — Москва: Дашков и К, Ай Пи Эр Медиа, 2023. — 211 с. — ISBN 978-5-394-05121-0.
2. Бондарь, А.Г. MS SQL Server 2022 / А.Г. Бондарь. — Санкт-Петербург: bhv, 2022. — 528 с. — ISBN 978-5-9775-1805-5.
3. Брайан Хоган HTML5 и CSS3. Веб-разработка по стандартам нового поколения / Хоган Брайан. — Санкт-Петербург : Питер, 2019. — 198 с.
4. Виафоре, П. Надежный Python / П. Виафоре. — БХВ, 2023. — 352 с. — ISBN 978-5-9775-1174-2.
5. Диков А.В. Web-программирование на стороне клиента: учебное пособие для бакалавров / Диков А.В. — Москва: Ай Пи Ар Медиа, 2022. — 461 с. — ISBN 978-5-4497-1629-3.
6. Кельчевская Н.Р. Бизнес-процессы промышленного предприятия: учебное пособие / Н.Р. Кельчевская, С.А. Сироткин, И.С. Пелымская [и др.] — Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2016. — 340 с. — ISBN 978-5-7996-1824-7.
7. Мухина Ю.Р. Web-дизайн: основы верстки сайтов: учебное пособие для СПО / Мухина Ю.Р. — Москва: Ай Пи Ар Медиа, 2023. — 155 с. — ISBN 978-5-4497-1790-0.
8. Павличева, Е. Н. Введение в информационные системы управления предприятием: учебное пособие / Е. Н. Павличева, В. А. Дикарев. — Москва: Московский городской педагогический университет, 2013. — 84 с. — ISBN 2227-8397.
9. Петкович, Д. Microsoft SQL Server 2012. Руководство для начинающих / Д. Петкович. — Пер. с англ. — Санкт-Петербург: БХВ-Петербург, 2013. — 816 с. — ISBN 978-5-9775-0854-4.

10. Прохоренок Н. А. Bootstrap и CSS-препроцессор Sass. Самое необходимое / Н. А. Прохоренок. — Санкт-Петербург: БХВ, 2021. — 496 с. — ISBN 978-5-9775-6769-5.
11. Флэнаган Д. JavaScript. Полное руководство / Д. Флэнаган. — ДиалектикаВильямс, 2021. — 720 с. — ISBN 978-5-907203-79-2.
12. Шенталер Ф. Бизнес-процессы: языки моделирования, методы, инструменты / Ф. Шенталер и др. — Москва: Альпина Паблишер, 2019. — 264 с. — ISBN 978-5-9614-2022-7.
13. Современный учебник JavaScript [Электронный ресурс] URL: <https://learn.javascript.ru/> (дата обращения: 30.04.2025).
14. Documentation // Bootstrap [Электронный ресурс] URL: <https://getbootstrap.com/> (дата обращения: 16.05.2025).
15. Documentation // Bootstrap 5 [Электронный ресурс] URL: <https://bootstrap5.ru/> (дата обращения: 23.05.2025).
16. Documentation // Django Packages [Электронный ресурс] URL: <https://djangopackages.org> (дата обращения: 20.05.2025).
17. Documentation // Django [Электронный ресурс] URL: <https://docs.djangoproject.com/en/4.1/topics/migrations/> (дата обращения: 16.05.2025).
18. Documentation // Django REST framework [Электронный ресурс] URL: <https://www.django-rest-framework.org/topics/documenting-your-api/> (дата обращения: 15.02.2025).
19. Documentation // Python [Электронный ресурс] URL: <https://docs.python.org> (дата обращения: 2.05.2025).
20. Mozilla Developer Network // Web technologies documentation [Электронный ресурс] URL: <https://developer.mozilla.org/> (дата обращения: 12.02.2025).

ПРИЛОЖЕНИЕ А
(обязательное)
КОД ПРЕДСТАВЛЕНИЙ

А.1 Представление для отображения договоров с истекшим сроком хранения

```
SELECT DISTINCT dz2.договор_ID_основной, dfs.файл_ID, dfs.документ_ID,
fc.код_доступа, dfs.имя_файла, dz2.[дата_закрытия]
FROM [dbo].[Договор_закрытие_2] AS dz2
INNER JOIN [dbo].[Договор] AS d
ON d.договор_ID_основной = dz2.договор_ID_основной
AND d.[дата_удаления] IS NULL
INNER JOIN [dbo].[Договор_FileStore] AS dfs
ON d.документ_ID = dfs.документ_ID
LEFT JOIN FileStore.dbo.файл_карточка_test AS fc
ON dfs.файл_ID = fc.файл_ID
WHERE DATEADD(year, 5, DATEADD(YEAR, DATEDIFF(YEAR, 0,
dz2.[дата_закрытия]) + 1, 0)) <= GETDATE()
AND NOT EXISTS(SELECT 1 FROM [dbo].._deleted_docs_list_of_contracts
ddlс WHERE ddlс.договор_ID_основной = dz2.договор_ID_основной)
```

А.2 Представление для просмотра договоров с истекшим сроком хранения,
которые не внесены в реестр

```
with contragent (предприятие_ID, ИНН, КПП, наименование_предприятия,
город, адрес_мс49, код_росатома) as
(select предприятие_ID, ИНН, КПП, наименование_предприятия, город,
адрес_мс49, код_росатома from _предприятие)
```

```
SELECT DISTINCT
d.договор_ID_основной, d.номер_договора, max(d.дата_договора) AS
дата_договора, CASE WHEN d.подразделение_ведущее_договор = '081' THEN
p.наименование_предприятия + ', ' + ISNULL(p.адрес_мс49, p.город) ELSE
p.город + ', ' + p.наименование_предприятия END AS город_поставщик
, d.кто_платит, max(total_sum) as сумма_договора_с_ндс
, d.валюта_договора, dd.дата_закрытия, d.регистрация_089

FROM [dbo].._del_docs dd
inner join dbo.Договор d on d.договор_ID_основной =
dd.договор_ID_основной
INNER JOIN contragent p ON p.предприятие_ID =
d.контрагент_предприятие_ID
INNER JOIN
(SELECT d.договор_ID_основной
, SUM(д.сумма_спецификации_без_ндс) + SUM(д.сумма_спецификации_ндс)
AS total_sum
FROM dbo.Договор d
```

```

GROUP BY д.договор_ID_основной)
AS sub ON d.договор_ID_основной = sub.договор_ID_основной
WHERE d.дата_удаления IS NULL

```

```

GROUP BY
д.договор_ID_основной, д.номер_договора, д.дата_договора, д.подразделение
_ведущее_договор, р.наименование_предприятия, р.адрес_мс49, р.город, р.гор
од, р.наименование_предприятия, д.кто_платит, д.валюта_договора, dd.дата_з
акрытия, д.регистрация_089, д.дата_удаления, dds.регистрация_089

```

А.3 Представление для просмотра договоров с истекшим сроком хранения, которые внесены в реестр

```

with contragent (предприятие_ID, ИНН, КПП, наименование_предприятия,
город, адрес_мс49, код_росатома) as
(select предприятие_ID, ИНН, КПП, наименование_предприятия,
город, адрес_мс49, код_росатома from _предприятие)

```

```

SELECT DISTINCT
д.договор_ID_основной, д.номер_договора, max(д.дата_договора) AS
дата_договора, CASE WHEN д.подразделение_ведущее_договор = '081' THEN
р.наименование_предприятия + ', ' + ISNULL(р.адрес_мс49, р.город) ELSE
р.город + ', ' + р.наименование_предприятия END AS город_поставщик
, д.кто_платит, max(total_sum) as сумма_договора_с_ндс
, д.валюта_договора, dd.dels_date_close, д.регистрация_089
, CASE WHEN dd.dels_date_del IS NOT NULL THEN CAST (1 AS BIT) ELSE
CAST(0 AS BIT) END AS is_dels_date_del

```

```

FROM [dbo].deleted_docs dd
inner join dbo.Договор d on д.договор_ID_основной =
dd.dels_contract_id_main
INNER JOIN contragent p ON p.предприятие_ID =
д.контрагент_предприятие_ID
INNER JOIN
(SELECT д.договор_ID_основной
, SUM(д.сумма_спецификации_без_ндс) + SUM(д.сумма_спецификации_ндс)
AS total_sum
FROM dbo.Договор д
GROUP BY д.договор_ID_основной) AS sub ON д.договор_ID_основной =
sub.договор_ID_основной
WHERE д.дата_удаления IS NULL

```

```

GROUP BY
д.договор_ID_основной, д.номер_договора, д.дата_договора, д.подразделение
_ведущее_договор, р.наименование_предприятия, р.адрес_мс49, р.город, р.гор
од, р.наименование_предприятия, д.кто_платит, д.валюта_договора, dd.dels_d
ate_close, dd.dels_date_del, д.регистрация_089, д.дата_удаления

```

А.4 Представление для просмотра договоров, направленных на закрытие

```

with E1_Arh (документ_ID) as
    (sSELECT DISTINCT документ_ID from dbo.Договор_FileStore
    where дата_удаления IS NULL),
    type_dok (тип_документа) as
    (select тип_документа from [dbo].[спр_Тип_документа] q
    where видимость = '%' union all
    select тип_документа from [dbo].[спр_Тип_документа] q
    where видимость in (select подразделение from
    dbo.[_исполнитель_текущий]))
SELECT DISTINCT
д.договор_ID_основной,д.документ_ID
,dbo._ф_статус_договора(д.договор_ID_основной) as статус_договора
,д.кто_платит
,номер_договора = case when ltrim(ISNULL(д.номер_договора, '')) = ''
and д.подразделение_ведущее_договор = '057' then д.номер_заявки else
д.номер_договора end,д.дата_договора
,дата_документа = case when ltrim(ISNULL(д.номер_договора, '')) = ''
and д.подразделение_ведущее_договор = '057' then д.дата_заявки else
д.дата_документа end
,исполнитель_договора = д.куратор_договора
,CASE WHEN д.подразделение_ведущее_договор = '081'
    THEN п.наименование_предприятия + ', ' + ISNULL(п.адрес_мс49,
п.город)
    ELSE п.город + ', ' + п.наименование_предприятия + ' (ИНН ' +
ISNULL(п.ИНН, '') + '; КПП ' + ISNULL(п.КПП, '') + ')'
    END AS город_поставщик
,qqq.валюта_договора,qqq.сумма_договора_без_ндс
,qqq.сумма_договора_ндс,qqq.сумма_договора_без_ндс +
qqq.сумма_договора_ндс as сумма_договора_с_ндс
,ISNULL(дсс.РОСАТОМ_согласованная_сумма, 0) as
РОСАТОМ_согласованная_сумма,rrr.регистрация_089,п.предприятие_ID
,ISNULL(post.нарушение_сроков_поставки, 0) as
нарушение_сроков_поставки,ISNULL(post.претензия_по_качеству, 0) as
претензия_по_качеству,эл_архив_договора = case when е1.документ_ID IS
NOT NULL THEN 'V' else NULL end,sum(количество_отправленных_писем) AS
letters,max([дата_отправки_письма]) max_дата_отправки_письма

FROM dbo.Договор д
inner join type_dok t on д.тип_документа = t.тип_документа
INNER JOIN dbo.[_предприятие] п ON п.предприятие_ID =
д.контрагент_предприятие_ID
INNER JOIN dbo.Журнал_отправки_писем_по_закрытию_договора j ON
j.договор_ID_основной = д.договор_ID_основной
LEFT JOIN
    (SELECT д.договор_ID_основной, д.валюта_договора,
    SUM(case when д.документ_ID is NULL THEN 0 ELSE 1 END) as
эл_согласование,
    SUM(д.сумма_спецификации_без_ндс) AS сумма_договора_без_ндс,
    SUM(д.сумма_спецификации_ндс) AS сумма_договора_ндс,

```



```

SUM(ROUND((ISNULL(д.курс_валюты_на_момент_заключения_договора,
1) * ISNULL(д.сумма_спецификации_без_ндс, 0)), 2)) as
сумма_договора_без_ндс_р,
SUM(ROUND((ISNULL(д.курс_валюты_на_момент_заключения_договора,
1) * ISNULL(д.сумма_спецификации_ндс, 0)), 2)) as сумма_договора_ндс_р
FROM dbo.Договор д
WHERE д.признак_суммирования = 0
AND д.дата_удаления IS NULL
GROUP BY д.договор_ID_основной, д.валюта_договора
) qqq on qqq.договор_ID_основной = д.договор_ID_основной
and qqq.валюта_договора = д.валюта_договора
LEFT JOIN
(select SUM(ISNULL(сумма_согласованная, 0)) as
POCATOM_согласованная_сумма, договор_ID_основной
from dbo.Договор_согласование_с_POCATOM
where дата_подтверждения_089 is NOT NULL
group by договор_ID_основной
) as дсс ON д.договор_ID_основной = дсс.договор_ID_основной
LEFT JOIN
(SELECT договор_ID_основной, MAX(дата_ввода) as дата_последняя,
регистрация_089
FROM dbo.Договор р WHERE дата_удаления IS NULL AND
регистрация_089 is not NULL GROUP BY договор_ID_основной,
регистрация_089
) rrr on rrr.договор_ID_основной = д.договор_ID_основной
LEFT JOIN dbo.Договор_оценка_поставщика post ON
д.договор_ID_основной = post.договор_ID_основной and
post.дата_удаления is NULL
LEFT JOIN El_Arh el ON д.электронный_архив_ID = el.документ_ID

where дата_документа >= '01.01.2022' and
д.подразделение_ведущее_договор not in ('057', '073') and
д.тип_документа not in ('Без договора', 'Без договора, оплачено
наличными', 'Упрощенный договор', 'Прочие документы') and
дата_окончания < getdate() and
dbo._ф_статус_договора(д.договор_ID_основной) not in ('Договор
Закрит', 'Договор Направлен на закрытие') and
д.признак_последнего_доп_соглашения IS NOT NULL AND д.дата_удаления IS
NULL

GROUP BY
д.договор_ID_основной, д.документ_ID, д.кто_платит, д.номер_договора, д.по
дразделение_ведущее_договор, д.номер_заявки, д.дата_договора, д.дата_заяв
ки, д.дата_документа, д.куратор_договора, п.наименование_предприятия, п.ад
рес_мс49, п.город, п.город, п.наименование_предприятия, п.ИНН, п.КПП, qqq.ва
люта_договора, qqq.сумма_договора_без_ндс, qqq.сумма_договора_ндс, дсс.PO
CATOM_согласованная_сумма, rrr.регистрация_089, п.предприятие_ID, post.на
рушение_сроков_поставки, post.претензия_по_качеству, el.документ_ID

```

ПРИЛОЖЕНИЕ Б

(обязательное)

КОД ПРОЦЕДУР

Б.1 Процедура для удаления связей между документом и файлами

```
CREATE PROCEDURE dbo._файл_задача_связь_del
    @файл_ID BIGINT,
    @задача_ID INT,
    @документ_ID BIGINT,
    @код_доступа VARCHAR(50)
AS
BEGIN
    -- Коды возврата:
    -- 1 - не найдена карточка на файл (может быть и неверный код доступа)
    -- 2 - есть другие ссылки на файл

    SET NOCOUNT ON
    SET XACT_ABORT ON

    DECLARE
        @первичный BIT
        , @tc INT
        , @cnt INT

    SELECT
        @первичный = фзс.[первичный]
    FROM [dbo].[файл_задача_связь] фзс
    INNER JOIN [dbo].[файл_карточка] фк ON фк.[файл_ID] = фзс.[файл_ID]
    AND фк.[код_доступа] = @код_доступа
    WHERE фзс.[файл_ID] = @файл_ID
        AND фзс.[задача_ID] = @задача_ID
        AND фзс.[документ_ID] = @документ_ID

    IF @первичный IS NULL RETURN 1

    SET @tc = @@TRANCOUNT

    IF @tc = 0 BEGIN TRAN
        -- Удаляем все ссылки данного документа на файлы
        DELETE FROM [dbo].[файл_задача_связь]
        WHERE [файл_ID] = @файл_ID
            AND [задача_ID] = @задача_ID
            AND [документ_ID] = @документ_ID

        -- Если не осталось ссылок на файл, то удаляем карточку файла и
        -- данные.
        DELETE [dbo].[файл_карточка]
        FROM [dbo].[файл_карточка] фк
```

```

LEFT JOIN [dbo].[файл_задача_связь] фзс ON фзс.[файл_ID] =
фк.[файл_ID]
WHERE фк.[файл_ID] = @файл_ID
AND фзс.[файл_ID] IS NULL

SET @cnt = @@ROWCOUNT

IF @cnt = 0 AND @первичный = 1 --Если ничего не удалили и связь
первичная,
--то отменяем удаление ее
BEGIN
    IF @tc = 0 ROLLBACK TRAN
    RETURN 2
END

IF @tc = 0 COMMIT TRAN
END
GO

```

Б.2 Процедура для получения содержательной части извещения о необходимости направления договора на закрытие

```

CREATE procedure dbo._мс49р_список_документов_для_закрытия_договоров
@s_xml varchar(max)
AS
BEGIN
    declare @idoc INTEGER, @RetCode INTEGER
    IF @s_xml IS NOT NULL
    BEGIN
        EXEC @RetCode = sp_xml_preparedocument @idoc OUTPUT, @s_xml
        IF @RetCode = 0
        BEGIN
            SELECT
                документ_ID_сэд, d.номер_документа, d.дата_документа,
d.текст_документа
                FROM OPENXML(@idoc, '/docIdMain/IdData', 0)
                WITH (документ_ID_сэд bigint) AS x
                INNER JOIN [dbo].[сэд_документ] d
                ON x.документ_ID_сэд = d.документ_ID ORDER BY
d.дата_документа
            END
            EXEC sp_xml_removedocument @idoc
        END
    END
END
GO

```

Б.3 Процедура для получения сообщений по конкретному извещению

```

CREATE procedure dbo._мс49р_список_переписка
@документ_id bigint
AS

```

```

SELECT
    адт.[адресат_ФИО_кратко] AS [автор], дпп.[сообщение_дата],
дпп.[сообщение_текст]
    ,дпп.[сообщение_общее], дпп.документ_ID, дпп.[сообщение_ID]
FROM [dbo].[сэд_документ_переписка] дпп
LEFT JOIN [dbo].[сэд_адресат] адт ON адт.[адресат_ID] =
дпп.[адресат_ID]
WHERE дпп.[документ_ID] = @документ_id AND дпп.[сообщение_общее] =
1
ORDER BY дпп.[сообщение_дата]
GO

```

Б.4 Процедура для удаления связей у всех файлов внутри таблицы

```

CREATE PROCEDURE dbo._del_link_to_deleted_docs
AS
DECLARE @задача_ID int
SET @задача_ID = 12
CREATE TABLE #t (dels_file_id BIGINT, dels_doc_id BIGINT,
dels_code_access NVARCHAR(50));
UPDATE dbo.deleted_docs
SET
    dels_date_del = GETDATE() -- 'YYYY-MM-DD hh:mm:ss[.nnn]'-
dels_date_del - datetime
OUTPUT INSERTED.dels_file_id, INSERTED.dels_doc_id,
INSERTED.dels_code_access
INTO #t (dels_file_id, dels_doc_id, dels_code_access)
WHERE dels_date_del IS NULL;
DECLARE @dels_file_id BIGINT, @dels_doc_id BIGINT, @dels_code_access
NVARCHAR(50)
DECLARE t_cursor CURSOR FOR
SELECT dels_file_id, dels_doc_id, dels_code_access FROM #t t
OPEN t_cursor
FETCH NEXT FROM t_cursor INTO @dels_file_id, @dels_doc_id,
@dels_code_access
WHILE @@fetch_status = 0
BEGIN
    EXECUTE FileStore.dbo._файл_задача_связь_del
        @dels_file_id,@задача_ID,@dels_doc_id,@dels_code_access
    UPDATE dbo.deleted_docs
        SET dels_date_del = GETDATE()
        WHERE dels_file_id = @dels_file_id AND dels_date_del IS NOT
NULL
    FETCH NEXT FROM t_cursor INTO @dels_file_id, @dels_doc_id,
@dels_code_access
END
CLOSE t_cursor
DEALLOCATE t_cursor
DROP TABLE #t
GO

```

ПРИЛОЖЕНИЕ В
(обязательное)
КОД ФАЙЛА views.py

В.1 Реализация пагинации

```
class listPaginator(PageNumberPagination):
    page_size = 20
    page_size_query_param = 'page_size'
    max_page_size = 1000
    def get_paginated_response(self, data):
        return super().get_paginated_response({
            "total_pages": self.page.paginator.num_pages,
            "page_num": self.page.number,
            "results": data
        })
```

В.2 Просмотр договоров, направленных на закрытие

```
class contractMainAdmViewSet(viewsets.ModelViewSet):
    queryset = ContractMainAdm.objects.all().order_by('-
max_date_send_letter')
    serializer_class = ContractMainAdmSerializer #(queryset,
many=True)
    pagination_class = listPaginator
    filter_backends = (DjangoFilterBackend, OrderingFilter)
    filterset_class = contractMainAdmFilter
    ordering_fields = ['max_date_send_letter', 'date_doc',
'sum_contract_without_tax', 'sum_contract_tax',
'sum_contract_with_tax', 'rosatom_agreed_sum', 'enterprise_id',
'letters', 'contravention_period_supply', 'claim_by_quality']
```

В.3 Просмотр договоров с истекшим сроком хранения, которые внесены в реестр

```
class deletedDocsListOfContractsViewSet(viewsets.ModelViewSet):
    queryset = DeletedDocsListOfContracts.objects.all().order_by('-
dels_date_close')
    serializer_class = DeletedDocsListOfContractsSerializer
    pagination_class = listPaginator
    filter_backends = (DjangoFilterBackend, OrderingFilter)
    filterset_class = DeletedDocsListOfContractsFilter
    ordering_fields = ['sum_contract_with_tax', 'dels_date_close',
'date_contract']
```

В.4 Просмотр договоров с истекшим сроком хранения, которые не внесены в реестр

```
class delDocsListOfContractsViewSet(viewsets.ModelViewSet):
    queryset = DelDocsListOfContracts.objects.all().order_by('-
date_closing')
    serializer_class = DelDocsListOfContractsSerializer
    pagination_class = listPaginator
    filter_backends = (DjangoFilterBackend, OrderingFilter)
    filterset_class = DelDocsListOfContractsFilter
    ordering_fields = ['sum_contract_with_tax', 'date_closing',
'date_contract']
```

В.5 Функция для получения извещений по конкретному договору

```
def sendDocIdSed(request):
    main_id = request.GET.get('contract_id_main')
    sed_ids =
JourneySendMailsClosing.objects.filter(contract_id_main=main_id).value
s_list('doc_id_sed', flat=True)
    ids = ""
    for i in sed_ids:
        ids += f'<IdData документ_ID_сэд= "{str(i)}" />'
    xml_data = f'<?xml version="1.0" encoding="WINDOWS-1251"
standalone="yes"?><docIdMain договор_ID_основной= "{str(main_id)}">' +
ids + '</docIdMain>'
    letters_sql = '''
        DECLARE @RC int
        DECLARE @s_xml varchar(max)
        SET @s_xml = %s
        exec @RC = dbo._mc49p_список_документов_для_закрытия_договоров
@s_xml
    '''
    result = MC49PListDocuments.objects.raw(letters_sql, [xml_data])
    json_letters = []
    for i in list(result):
        json_letters.append({
            "id_sed": i.id_sed,
            "doc_no": i.doc_no,
            "date_doc": i.date_doc,
            "text_doc": i.text_doc
        })
    return JsonResponse ({'letters': json_letters}, safe=False)
```

ПРИЛОЖЕНИЕ Г
(обязательное)
КОД ФАЙЛА serializers.py

Г.1 Сериализатор договоров с истекшим сроком хранения, которые не внесены в реестр

```
class DelDocsListOfContractsSerializer(serializers.ModelSerializer):  
    class Meta:  
        model = DelDocsListOfContracts  
        fields = '__all__'
```

Г.2 Сериализатор договоров с истекшим сроком хранения, которые внесены в реестр

```
class  
DeletedDocsListOfContractsSerializer(serializers.ModelSerializer):  
    class Meta:  
        model = DeletedDocsListOfContracts  
        fields = '__all__'
```

Г.3 Сериализатор договоров, направленных на закрытие

```
class ContractMainAdmSerializer(serializers.ModelSerializer):  
    class Meta:  
        model = ContractMainAdm  
        fields = '__all__'
```

ПРИЛОЖЕНИЕ Д
(обязательное)
ПАГИНАЦИЯ ТАБЛИЦЫ

Код функционала на языке JavaScript:

```
function updatePagination(data){
    const paginationElement = document.getElementById('paginator');
    paginationElement.innerHTML = '';
    if(1 != data.results.page_num){
        const lastButn = document.createElement('button');
        lastButn.className = 'btn_color';
        lastButn.innerText = 'Первая';
        lastButn.onclick = () => loadPage(1);
        paginationElement.appendChild(lastButn);
    }
    if(data.previous){
        const prevButn = document.createElement('button');
        prevButn.className = 'btn_color';
        prevButn.innerText = 'Предыдущая';
        prevButn.onclick = () => loadPage(data.results.page_num - 1);
        paginationElement.appendChild(prevButn);
    }
    const currentPage = document.createElement('span');
    currentPage.innerText = ` ${data.results.page_num} `;
    paginationElement.appendChild(currentPage);
    if(data.next){
        const nextButn = document.createElement('button');
        nextButn.className = 'btn_color';
        nextButn.innerText = 'Следующая';
        nextButn.onclick = () => loadPage(data.results.page_num + 1);
        paginationElement.appendChild(nextButn);
    }
    if(data.results.total_pages != data.results.page_num){
        const lastButn = document.createElement('button');
        lastButn.className = 'btn_color';
        lastButn.innerText = data.results.total_pages;
        lastButn.onclick = () => loadPage(data.results.total_pages);
        paginationElement.appendChild(lastButn);
    }
    if(data.results.total_pages > 1){
        const inputPage = document.createElement('input');
        inputPage.className = `text_card`;
        inputPage.placeholder = ` Перейти на страницу `;
        inputPage.onchange = function() {
            if(1 <= inputPage.value && inputPage.value <=
data.results.total_pages)
                loadPage(inputPage.value);
            else alert(`Такой страницы не существует. Текущий диапазон: 1-
${data.results.total_pages}`)
        }
        paginationElement.appendChild(inputPage);
    }
};
```


ПРИЛОЖЕНИЕ Е

(обязательное)

НАВИГАЦИОННАЯ ПАНЕЛЬ

Е.1 HTML-разметка

```
<header class="heading">
  <nav class="heading_nav">
    <ul class="heading_ul" id="h_name_select" >
      {% for i in json_data.places %}
        {% if i.value == current_heading_name %}
          <li>
            <button class="heading_btn_active"
id="{{i.value}}" onclick="goToHeadind('{{contract_id_main}}',
'{{i.value}}')">{{i.title}}</button>
          </li>
        {% else %}
          <li>
            <button class="heading_btn"
id="{{i.value}}" onclick="goToHeadind('{{contract_id_main}}',
'{{i.value}}')">{{i.title}}</button>
          </li>
        {% endif %}
      {% endfor %}
    </ul>
  </nav>
</header>
```

Е.2 CSS стили

```
.heading{
  margin:0;
  display:flex;
  flex-direction: column;
  justify-content: flex-end;
  align-items: center;
}
.heading_nav{
  background: rgb(239 244 249);
  border-radius: 16px 16px 0 0;
  position: fixed;
  bottom: 0;
  display: flex;
  justify-content: space-around;
  padding: 0;
  width: auto;
  align-items: stretch;
  height:40px
```

```

}
.heading_ul{
    display: flex;
    margin-bottom: 0;
    list-style: none;
    justify-content: center;
    padding-right: 28px;
}
.heading_btn{
    border: none;
    background: rgb(239 244 249);
    flex: 1;
    display: flex;
    align-items: center;
    justify-content: center;
    padding: 10px;
}
.heading_btn_active{
    border: none;
    background: rgb(222, 231, 240);
    flex: 1;
    display: flex;
    align-items: center;
    justify-content: center;
    padding: 10px;
}
.heading_btn:hover{
    background: rgba(247, 249, 255, 0.5);
}

```