

УТВЕРЖДАЮ

Генеральный директор
ООО «ЦНЦ «БизнесАвтоматика»

П.С. Петраков

2023 г.

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА «ВИЗАРИ СЭД»

Детализированное техническое задание, включая функциональные требования, требования к структуре и функционированию Решения, требования к показателям надежности, требования к эргономике и технической эстетике, требования к спецификации и унификации, требования к лингвистическому обеспечению, требования к техническим средствам, утвержденное генеральным директором

Москва, 2023



Оглавление

Термины и сокращения	4
1 Общие сведения	7
1.1 О документе	7
1.2 Полное наименование	7
1.3 Перечень документов, на основании которых ведется разработка Решения.....	7
1.4 Сроки выполнения работы.....	7
2 Назначение и цели разработки Решения	8
2.1 Назначение Решения.....	8
2.2 Цели создания Решения.....	8
3 Характеристика объекта автоматизации	10
4 Требования к Решению	11
4.1 Требования к Решению в целом	11
4.1.1 Требования к структуре и функционированию Решения	11
4.1.2 Требования к численности и квалификации персонала Решения и режиму его работы	17
4.1.3 Требования к показателям назначения	18
4.1.4 Требования безопасности.....	19
4.1.5 Требования к эргономике и технической эстетике	19
4.1.6 Требования к показателям надежности	21
4.1.7 Требования по стандартизации и унификации	23
4.1.8 Требования к защите информации.....	24
4.2 Требования к формированию детализированных функциональных требований к разработке Решения.....	25
4.3 Функциональные требования к компонентам (модулям) Решения.....	25
4.3.1 Функциональные требования к компоненту (модулю) электронного документооборота.....	25
4.3.2 Функциональные требования к компоненту нормативно-справочной информации	34
4.3.3 Функциональные требования к интеграционной шине	35
4.3.4 Функциональные требования к компоненту (модулю) предобработки информации ETL	38
4.3.5 Функциональные требования к компоненту (модулю) автоматизированного проектирования предметной области	40
4.3.6 Функциональные требования к компоненту (модулю) защищенного распределенного хранения данных с встроенными редакторами документов	43
4.3.7 Функциональные требования к компоненту (модулю) управления регламентированными процессами.....	44
4.3.8 Функциональные требования к компоненту (модулю) управления задачами	47
4.3.9 Функциональные требования к компоненту (модулю) уведомлений	53
4.3.10 Функциональные требования к компоненту (модулю) поиска.....	54
4.3.11 Функциональные требования к компоненту (модулю) безопасности.....	56
4.3.12 Функциональные требования к компоненту (модулю) администрирования	57
4.4 Требования к видам обеспечения Решения.....	58
4.4.1 Требования к математическому обеспечению.....	58
4.4.2 Требования к информационному обеспечению.....	58
4.4.3 Требования к программному обеспечению.....	59
4.4.4 Требования к лингвистическому обеспечению	59
4.4.5 Требования к техническим средствам	60
5 Состав и содержание работ по разработке Решения	61
5.1 Требования к детализированной модели монетизации Решения	61
5.2 Требования расчетным алгоритмам, используемым для Решения	61
5.3 Порядок разработки и этапы работ по разработке Решения.....	62
5.4 Требования к организациям-соисполнителям работ	64
6 Порядок контроля и приемки результатов работ по разработке Решения	65

6.1 Виды, состав и методы испытаний разработанного Решения и его составных частей	65
6.1.1 Требования к видам испытаний Решения	65
7 Требования к документированию	66
7.1 Перечень документов, предъявляемых по окончании соответствующих этапов работ	66
8 Требования к регистрации ПО Решения	67
9 Источники разработки	68

Термины и сокращения

В настоящем документе применены следующие термины с соответствующими определениями, представленные в таблице 1:

Таблица 1 – Термины и определения

Термин	Определение
1	2
Исполнитель	ООО «Научно-производственный центр «БизнесАвтоматика», НПЦ «БизнесАвтоматика»
Проект	Комплекс мероприятий, направленных на разработку Решения
Решение	Автоматизированная информационная система электронного документооборота «Визари СЭД»
Визари	Программное решение, разработанное специалистами НПЦ «БизнесАвтоматика, зарегистрированное в Едином реестре российских программ для электронных машин и баз данных (№6653 от 23.04.2020 г.).
BigData	(от англ. Big Data) – обозначение структурированных и неструктурированных данных значительного объема, обработка и анализ которых требует использования современных программных решений
Deep learning	(от англ. Deep Learning) – совокупность методов машинного обучения, основанных на обучении представлением, а не специализированных алгоритмах под конкретные задачи
WorkFlow	(от англ. Workflow) – графическое представление потока задач в рамках процессов, подпроцессов работ

В настоящем документе применены следующие сокращения с соответствующими определениями, представленные в таблице 2:

Таблица 2 – Сокращения и определения

Сокращение	Определение
1	2
АИС	Автоматизированная информационная система
БП	Бизнес-процесс
ВПСНС	Визуальное проектирование сверточных нейронных сетей
ЕСИА	Единая система идентификации и аутентификации
ИИ	Искусственный интеллект
ИМ	Информационная модель
ИНС	Искусственная нейронная сеть
ИТ	Информационные технологии
НДС	Налог на добавленную стоимость
НСИ	Нормативно-справочная информация
ОС	Операционная система
ПИ РНС	Программное изделие разработки нейронных сетей
ПИ СОД	Программное изделие синтеза обучающих данных
ПО	Программное обеспечение
СМЭВ	Система межведомственного электронного взаимодействия
СОА	Сервисно-ориентированная архитектура
СОВРСД	Синтез обучающей выборки и разметки синтезированных данных
СПО	Специальное программное обеспечение

Сокращение	Определение
1	2
СУБД	Система управления базой данных
ЭВМ	Электронная вычислительная машина
AI	(от англ. Artificial Intelligence) – комплекс технологий, на базе которой происходит разработка интеллектуальных компьютерных систем
API	(от англ. Application Programming Interface) – набор способов и правил, по которым информационные системы производят обмен данными
BI	(от англ. Business Intelligence) – программные решения, основная функциональная задача которых заключается в обработке и визуализации результатов анализа различной информации
BPM	(от англ. Business Process Management) – управление бизнес-процессами, концепция процессного управления организацией, рассматривающая бизнес-процессы как особые ресурсы предприятия
CSV	(от англ. Comma-Separated Values) – текстовый формат, предназначенный для представления табличных данных. Строка таблицы соответствует строке текста, которая содержит одно или несколько полей, разделенных запятыми, или точками с запятой
CSS	(от англ. Cascading Style Sheets) – каскадные таблицы стилей, формальный язык описания внешнего вида веб-страницы, написанного с использованием языка разметки
ETL	(от англ. Extract, Transform, Load) – инструмент управления данными, обеспечивающий возможность извлечения, преобразования, очистки и загрузки данных в хранилище данных
DL	(от англ. Deep Learning) — это вид машинного обучения с использованием многослойных нейронных сетей, которые самообучаются на большом наборе данных.
HTML	(от англ. HyperText Markup Language) – язык гипертекстовой разметки, стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере
HTTP	(от англ. HyperText Transfer Protocol) – протокол прикладного уровня передачи данных
JSON	(от англ. JavaScript Object Notation) – текстовый формат обмена данными
KPI	(от англ. Key Performance Indicator) – ключевой показатель эффективности, числовой показатель деятельности, позволяющий измерить степень достижения целей и оптимальности процесса
ML	(от англ. Machine Learning) – метод анализа данных, поддерживающий процесс обучения программного аналитического продукта в ходе обработки и анализа данных для решения новых задач.
MVC	(от англ. Model-View-Controller) – схема разделения данных приложения, и управляющей логики на три отдельных блока: модель, представление и контроллер
ODBC	(от англ. Open Database Connectivity) – программный интерфейс (API) доступа к базам данных

Сокращение	Определение
1	2
REST	(от англ. Representational State Transfer) – передача репрезентативного состояния, архитектурный стиль взаимодействия распределенного приложения в сети
SOAP	(от англ. Simple Object Access Protocol) – простой протокол доступа к объектам, протокол обмена структурированными сообщениями в распределенной вычислительной сети
SQL	(от англ. Structured Query Language) – язык структурированных запросов, декларативный язык программирования, применяемый для создания, модификации и управления данными реляционной базы данных
SSL	(от англ. Secure Sockets Layer) – уровень защищённых сокетов) – криптографический протокол, который подразумевает более безопасную связь
XML	(от англ. eXtensible Markup Language) – расширяемый язык разметки, предназначенный для хранения и передачи данных
XSLT	(от англ. (Extensible Stylesheet Language Transformations) – язык преобразования XML-документов

1 Общие сведения

1.1 О документе

Документ содержит детализированные функциональные и нефункциональные требования к автоматизированной информационной системе электронного документооборота «Визари СЭД» (далее также – Решение).

1.2 Полное наименование

Автоматизированная информационная система электронного документооборота «Визари СЭД» (АИС «Визари СЭД»).

1.3 Перечень документов, на основании которых ведется разработка Решения

- Детализированное описание модели монетизации Решения;
- Функциональные требования и спецификация программного обеспечения системы электронного документооборота «Визари СЭД».

1.4 Сроки выполнения работы

Начало работ: 16.01.2023

Окончание работ: 25.09.2023

2 Назначение и цели разработки Решения

2.1 Назначение Решения

Решение предназначено для автоматизации полного цикла процессов управленческого документооборота, делопроизводства и контроля исполнительской дисциплины в организациях различной отраслевой направленности и масштаба, включая государственные учреждения, банки, страховые компании, торговые сети и другие.

2.2 Цели создания Решения

Платформа «Визари» – это высокоэффективная клиент-серверная платформа, позволяющая автоматизировать любые организационные и технологические процессы организации.

Приложения, построенные на платформе «Визари», создаются по модульному принципу, что позволяет добавлять требуемый функционал с помощью выбора и настройки готовых модулей и подсистем. Итоговый набор подсистем и модулей может быть индивидуально подобран под особенности инфраструктуры предприятия любого масштаба. Разработка АИС «Визари СЭД» на платформе «Визари» позволит оптимизировать функционал системы в области процессов создания, обработки и управления документацией в предприятиях различного масштаба.

АИС «Визари СЭД» должна быть востребована среди предприятий различного масштаба, в том числе Федеральными органами государственной власти Российской Федерации, органами государственной власти и местного самоуправления субъектов Российской Федерации, предприятиями (организациями) государственного и корпоративного секторов, крупного, среднего и малого бизнеса Российской Федерации.

Регистрация Решения в Едином реестре российских программ для электронных вычислительных машин и баз данных сделает программное решение особенно востребованным среди государственных заказчиков, в том числе государственных корпораций, которые обязаны соблюдать требования Постановления Правительства Российской Федерации от 16.11.2015 № 1236 «Об установлении запрета на допуск программного обеспечения, происходящего из иностранных государств, для целей осуществления закупок для обеспечения государственных и муниципальных нужд».

Разработка Решения выполняется с целью создания современного, функционального, конкурентоспособного как на российском, так и на мировом рынке программного решения, предусматривающего применение технологий искусственного интеллекта, предоставляющего широкий спектр возможностей, упрощающих процесс подготовки и обработки документов, а также характеризующегося интуитивно понятным и удобным пользовательским интерфейсом, позволяющим использовать Решение без привлечения специалистов по данным или ИТ-специалистов.

Цели создания Решения:

- внедрение простого и понятного решения, не требующего значительных затрат на обучение сотрудников и внедрение;
- предоставление пользователям функционального «коробочного» (готового к эксплуатации без значительного объема доработок) решения, построенного на базе технологичной и гибкой системы;

– преодоление «геополитических» (формальные или неформальные) ограничений в возможностях использования программных продуктов от вендоров из определенных стран.

Объединение самостоятельных компонентов (модулей) Решения в рамках единой версии Решения обеспечит широкую востребованность на отечественном и международном рынке. Решение позволит заказчикам с минимальными затратами осуществлять переход от широко растиражированных иностранных продуктов (Documentarium, DOCS Open и DOCS Fusion, Staffware, Panagon, DocuLive, Lotus Notes), а также составит конкуренцию наиболее распространенным отечественным СЭД (БОСС-Референт, Кодекс: Документооборот, Гран-док, Ефрат, Дело, LanDocs, Крон, OfficeMedia, Effect Office, N.System, LS Flow, Оптима, ЭСКАДО, Тезис, 1С:Документооборот и 1С:Архив).

3 Характеристика объекта автоматизации

Объектом автоматизации является полный цикл документооборота и делопроизводства организаций различной отраслевой направленности и масштаба, включая государственные организации и территориально-распределенные организации.

4 Требования к Решению

4.1 Требования к Решению в целом

4.1.1 Требования к структуре и функционированию Решения

4.1.1.1 Решение должно быть ориентировано на централизованное обслуживание и обеспечение удаленного доступа к данным и возможность полнофункционального использования Решения через веб-браузер.

4.1.1.2 Для обеспечения возможностей по масштабируемости и изменению конфигурации Решения, а также безопасности и отказоустойчивости при доработке компонентов должна быть использована трехзвенная архитектура, состоящая из клиента, сервера приложений и сервера баз данных (рисунок 1).

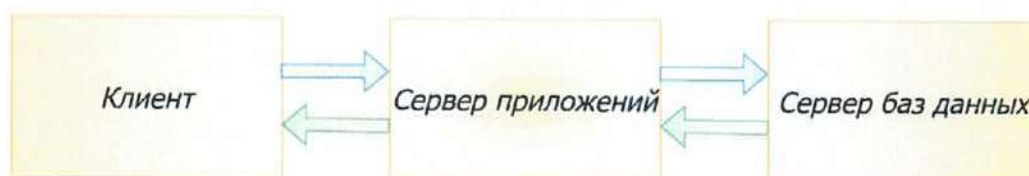


Рисунок 1 – Трехзвенная клиент-серверная архитектура

4.1.1.3 Логическая структура Решения должна содержать три уровня: хранения данных, бизнес-логики и взаимодействия с пользователями (презентационный).

- Презентационный уровень должен представлять данные пользователю, а также обеспечивать возможность ввода и изменения данных при наличии у пользователя соответствующих прав.

- Уровень бизнес-логики должен содержать программные объекты и программный код, которые реализуют логику работы задач. Компоненты должны выполняться на сервере и разделять общие ресурсы сервера.

- Уровень хранения данных должен обеспечивать долговременное эффективное хранение данных. Уровень хранения должен включать СУБД и компоненты для доступа к данным. Уровень хранения должен предоставлять программный интерфейс для объектов уровня бизнес-логики.

4.1.1.4 Решение должно предусматривать возможности дальнейшей модификации функциональных модулей на основе разработанных решений.

Должен быть использован открытый (open-source) стек-технологии: PostgreSQL, C#.

4.1.1.5 Должна быть предусмотрена возможность масштабирования Решения при увеличении нагрузки. При увеличении нагрузки должно обеспечиваться горизонтальное масштабирование системы – увеличение производительности приложения за счёт распределения нагрузки между оборудованием.

4.1.1.6 Решение должно быть спроектировано по модульному принципу (иметь микросервисную архитектуру) для исключения дублирования функциональности и обеспечения возможности обновления компонентов без негативного воздействия друг на друга.

4.1.1.7 Решение должно соответствовать следующим принципам:

- Открытость – Решение должно использовать общедоступные и специфицированные решения, протоколы и интерфейсы; открытость Решения должно позволять выполнить проверку исходных кодов на отсутствие ошибок; архитектура

Решение не должно ограничивать возможности по разработке новых функций и выбора Исполнителя, а также накладывать ограничения на масштабируемость инфраструктуры потенциального заказчика; архитектура Решения не должна привязывать ее к использованию конкретных проприетарных продуктов.

- Модульность – Решение должно быть построено с использованием модульной (микросервисной) архитектуры, состоящей из отдельных компонентов, обеспечивающих возможность их независимой модификации; сбой в работе одного из модулей или одного из компонентов не должен приводить к полному прекращению функционирования Решения в целом.

- Масштабируемость – архитектура Решения должна позволять увеличивать производительность, объемы хранимой и обрабатываемой информации при добавлении ресурсов за счёт распределения нагрузки между оборудованием.

- Управляемость и конфигурируемость – должно быть обеспечено управление элементами Решения на всех уровнях его архитектуры: на уровне инфраструктуры, на функциональном уровне, на уровне представления данных.

- Индивидуализация – должен быть обеспечен удобный индивидуальный доступ к Системе для всех потенциальных групп пользователей с функциональностью, соответствующей их задачам.

- Функциональное соответствие – архитектура Решения должна отвечать функциональным задачам Решения.

- Системность – все взаимосвязанные элементы Решения должны использовать единую методологию и отвечать единым принципам взаимодействия, надежности и управления.

- Единство графического представления – при проектировании и разработке пользовательских интерфейсов должны использоваться общие принципы графического представления информации и организации доступа пользователей к функциональным возможностям и сервисам компонентов.

- Управление содержанием и представлением информации – обеспечение эффективных методов создания, изменения, сохранения и удаления информационного содержания и структуры Решения, механизмов управления процессом создания и публикации документов Решения; обеспечение средств для создания и изменения представления информации в Решении; способность создания и контроля информационного наполнения Решения персоналом, не обладающим техническими знаниями, при помощи дружественного и интуитивно понятного интерфейса.

- Навигация и поиск – должны быть обеспечены средства навигации внутри Решения, поиска информации в Решении с использованием различных типов запросов, а также средства представления результатов поиска в удобной и изменяемой форме.

- Анализ использования информации – обеспечение механизмов эффективного сбора необходимой информации и анализа использования ресурсов Решения, данных о доступе к ресурсам Решения (например, тип страницы или время суток).

4.1.1.8 Решение должно иметь следующую микросервисную архитектуру:

- Компонент (модуль) электронного документооборота.
- Компонент (модуль) нормативно-справочной информации;
- Интеграционная шина;

- Компонент (модуль) предобработки информации ETL;
- Компонент (модуль) автоматизированного проектирования предметной области;
- Компонент (модуль) защищенного распределенного хранения данных с встроенными редакторами документов;
- Компонент (модуль) управления регламентированными процессами;
- Компонент (модуль) управления задачами;
- Компонент (модуль) уведомлений;
- Компонент (модуль) поиска;
- Компонент (модуль) безопасности;
- Компонент (модуль) администрирования.

4.1.1.9 Все компоненты Решения должны быть спроектированы по шаблону MVC (Model-View-Controller – «Модель-Представление-Контроллер») – схема разделения данных приложения, и управляющей логики на три отдельных блока: модель, представление и контроллер (рисунок 2). Такое разделение позволяет модифицировать каждый уровень независимо.

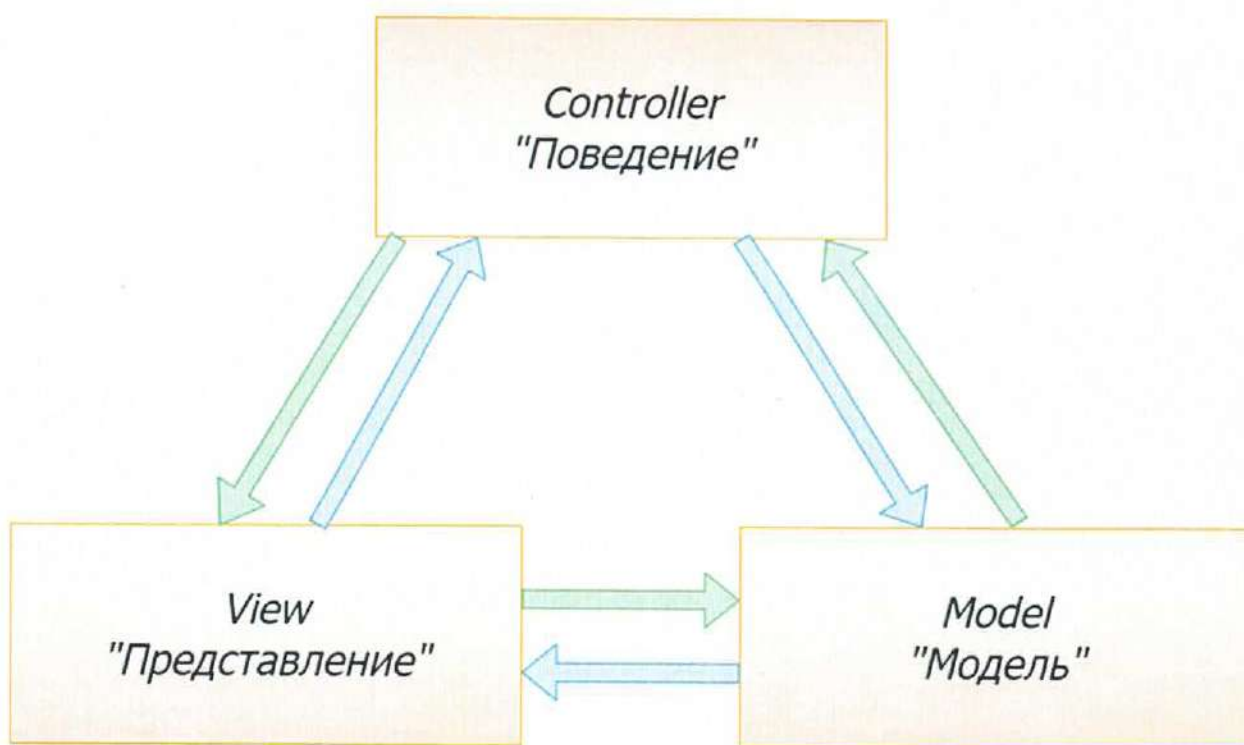


Рисунок 2 – Шаблон проектирования «Модель-Представление-Контроллер»

Блоки в шаблоне MVC выполняют следующие функции:

- «Модель» – реализует бизнес-логику приложения и выполняет обработку данных;
- «Контроллер» – обрабатывает запросы пользователя и выполняет запросы к другим компонентам;
- «Представление» – осуществляет визуализацию полученного результата или упаковывает полученный результат в необходимый формат.

Преимущества MVC:

- сокращение временных и материальных затрат на доработку и отладку сложных приложений корпоративного уровня;
- сокращение временных и материальных затрат на внедрение новых технологий (любой из уровней может быть обновлен без переработки остальных).

4.1.1.10 Платформа «Визари» построена на сервисно-ориентированной архитектуре (COA).

За счёт высокой эффективности масштабирования COA большая часть сервисов «Визари» собраны в программные высокопроизводительные и отказоустойчивые кластеры, готовые для использования в высоконагруженных информационных системах, например, программный кластер сервисов СУБД, сервисов файлового хранилища, сервисов анализа данных, интеграционных сервисов, геоинформационных сервисов, коммуникационных сервисов и т.д.

COA обладает следующими преимуществами:

- изолированность: при разворачивании сервисов получают изолированные функциональные блоки, которые не влияют на работоспособность друг друга;
- производительность: ядро с сильным связыванием позволяет взаимодействовать микросервисам через кэш центрального процессора или оперативную память, минуя накладные сетевые расходы;
- доступность: отсутствие обязательного требования к функционированию всех сервисов системы в любой момент времени;
- гибкость и гетерогенность: возможность реализации модулей системы на разных технологиях;
- независимость при развертывании: благодаря слабой связанности сервисов достигается простота разворачивания системы и минимизируется вероятность её отказа, достигается отказоустойчивость и надежность системы в целом;
- масштабируемость: возможность распределения сервисов между разными виртуальными или физическими вычислительными узлами.

4.1.1.11 Сервисно-ориентированная архитектура типового решения, построенного на платформе «Визари» включает следующий список сервисов:

- веб-сервис – сервис для публикации данных для внешних пользователей. Для работы используется сервер nginx с использованием технологии тонкого клиента (web). Может включать в себя сервис для взаимодействия с мобильными приложениями;
- сервисы Веб-приложений, включающие в себя:
 - сервис для обработки информации и взаимодействия между приложениями (graphql);
 - сервис авторизации пользователей и администраторов (identity);
 - сервис для хранения пользовательских данных (profile);
 - сервис для взаимодействия со сторонними сервисами (web-api);
 - базовые и прикладные подсистемы и модули платформы «Визари», например:
 - сервис для взаимодействия между пользователями (ikr);
 - геоинформационный сервис;
 - аналитический сервис;
 - сервис поиска;

– и т.д.

– Сервис Баз Данных позволяет использовать несколько видов баз данных для оптимального доступа к различным типам данных. Например, для хранения и обработки данных пользователей используется реляционная СУБД PostgreSQL/Postges Pro, а для данных основной сценарий работы с которыми предполагает частый поиск – NoSQL СУБД Redis и Elasticsearch (хранение, индексация, кеширование).

– Сервис хранения данных – сервис для предоставления доступа к виртуальному хранилищу данных (filestorage).

– Сервис очередей – сервис, который обеспечивает возможность взаимодействия между компонентами и сервисами (rabbitmq).

– Сервис Резервных копий – служебный сервис для периодического резервного копирования компонентов и данных системы. Резервируются файлы, базы данных, индексы Elasticsearch.

– Сервис Мониторинга – служебный сервис обеспечивающий мониторинг сервисов платформы «Визари». В состав сервиса входят компоненты — Elasticsearch, logstash, kibana, а также сервисы предоставления данных (beats).

– Служебные сервисы – служебные сервисы, включающие в себя сервис DNS, NTP, систему централизованного управления (ansible).

4.1.1.12 При повышении нагрузки должны создаваться несколько экземпляров сервисов, которые объединяются в кластеры и данные начинают распределяться между отдельными экземплярами внутри кластера. Пример таких кластеров – кластер Elasticsearch, кластер PostgreSQL.

4.1.1.13 Требования к способам и средствам связи между компонентами Решения
Информационный обмен между серверными компонентами Решения и клиентскими приложениями должен осуществляться по сети Интернет, посредством протокола HTTP.

Взаимосвязь между компонентами Решения должна быть организована на уровне БД при помощи механизма внешних ключей, либо на уровне программного кода.

Взаимодействие подсистем должно обеспечить консистентность данных по одним и тем же объектам вне зависимости от способа их представления.

4.1.1.14 Требования к режимам функционирования Решения

Решение должно поддерживать следующие режимы функционирования:

- штатный режим работы;
- режим технического обслуживания (обновления отдельных компонентов программно-технического комплекса);
- режим администрирования;
- режим аварийного завершения работы;
- режим восстановления работы после аварии;
- режим работы с частичной потерей функциональности.

При условии регулярного регламентного обслуживания и мониторинга параметров работы Решения должна обеспечить длительно-непрерывное, круглосуточное функционирование в штатном режиме и в режиме системного администрирования.

Основным режимом функционирования должен являться штатный режим, при котором Решение должно поддерживать выполнение всех заявленных функций. В этом

режиме должна быть обеспечена работа всех зарегистрированных пользователей в круглосуточном режиме.

Режим технического обслуживания предназначен для проведения запланированных работ по обслуживанию Решения и может сопровождаться частичной недоступностью функциональности.

Обновления отдельных компонентов Решения должны проводиться в фоновом режиме без прекращения работы пользователей.

В режиме администрирования Решение должно обеспечивать возможность проведения следующих работ:

- настройка среды функционирования;
- пополнение программных средств новыми компонентами;
- мониторинг, контроль и диагностирование работоспособности;
- копирование и архивирование баз данных;
- копирование и архивирование прикладных программных компонент;
- импорт и экспорт данных для обмена с внешними системами.

Должна быть предоставлена возможность настройки определенного временного периода для автоматического сохранения внесенных в Систему изменений.

Система переходит в аварийный режим при возникновении нештатной ситуации и невозможности штатной работы. Режим аварийного завершения производится с предварительным автоматическим уведомлением пользователей и предложением завершить работу с Системой. Аварийный режим функционирования характеризуется отказом одного или нескольких компонентов Решения. В этом режиме должны приниматься меры к обеспечению временной работоспособности Решения (возможно с ограничением в выполнении отдельных функций), после чего проводится анализ причин выхода Решения в аварийный режим и проводятся мероприятия по восстановлению полной работоспособности.

Режим восстановления работы после аварии выполняется в автоматическом режиме с переключением на резервную копию Системы в случае ее наличия.

Режим работы с частичной потерей функциональности в основном предназначен для:

- проведения реконфигурирования;
- профилактического обслуживания;
- регламентированного обновления и информационного взаимодействия с источниками данных, не предполагающих непрерывную актуализацию.

4.1.1.15 Требования по диагностированию Решения

Решение должно быть снабжена средствами диагностики работоспособности. В случае выявления проблем, сведения о них должны сообщаться пользователю и помещаться в протокол сбоев и ошибок.

Решение должно гарантировать отсутствие потерь и искажений информации.

Диагностика работоспособности программного обеспечения, реализующего интерфейсы пользователей, должна осуществляться стандартными средствами клиентских операционных систем, управляющих функционированием рабочих станций.

Диагностирование Решения должно осуществляться путем анализа записей в системных журналах СУБД, веб-сервера и операционной системы, а также с помощью встроенных средств диагностирования общего программного обеспечения Системы.

В Решения диагностированию подлежат:

- отклонения от заданных параметров быстрогодействия Решения;
- случаи аварийных остановок и самопроизвольной перезагрузки программного обеспечения;
- случаи нарушений целостности баз данных и файловой системы;
- сбои при выполнении регламентных операций резервного копирования;
- пользовательские действия и запросы;
- доступность технологических интерфейсов интеграции Решения с внешними информационными системами;
- функционирование модулей (компонентов) Решения;
- осуществление резервного копирования и обеспечения безопасности.

4.1.1.16 Перспективы развития, модернизации Решения

Решение должно предусматривать возможности дальнейшей модификации функционала, работ и сервисов на основе разработанных решений. Должна быть предусмотрена возможность масштабирования Решения при увеличении нагрузки. Решение должно быть спроектировано по модульному принципу для исключения дублирования функционала, а также возможности обновления модулей без негативного воздействия друг на друга.

4.1.2 Требования к численности и квалификации персонала Решения и режиму его работы

Численность персонала определяется количеством пользователей Решения.

Пользователи должны обладать навыками работы на компьютере.

Администраторы Решения должны обладать навыками администрирования информационных систем.

Администраторы модулей (компонентов) Решения должны обеспечивать функционирование модулей (компонентов), сохранность данных, их квалификация должна позволять:

- ориентироваться в стандартных возможностях используемых операционных систем и СУБД, протоколах передачи данных;
- владеть средствами мониторинга СУБД;
- владеть средствами резервного копирования и восстановления данных;
- владеть средствами защиты информации;
- работать с архиваторами, дисковыми утилитами, антивирусными программами;
- определять источник сбоя функционирования ПО и грамотно описывать его.

Режим работы персонала должен соответствовать режиму работы предприятия.

Список типовых ролей Решения представлен в виде матрицы доступа в таблице 3.

Таблица 3 – Матрица доступа

Роль/группа	Уровень доступа к данным	Разрешенные действия
Администратор Решения	Обладает полной информацией о системном и прикладном программном обеспечении Решения. Обладает полной информацией о технических средствах и конфигурации Решения. Имеет доступ ко всем техническим средствам обработки информации и данным Решения. Имеет доступ к средствам защиты информации и протоколирования и к части ключевых элементов Решения. Обладает правами конфигурирования и административной настройки технических средств Решения	Добавление пользователей. Редактирование личных данных пользователя. Удаление пользователей. Настройка резервного копирования. Рассылка массовых уведомлений пользователям. Включение /отключение функции управления рабочим столом. Включение /отключение автоматической рассылки уведомлений. Настройка портов почтового сервера. Настройка параметров под бета-версию Решения безопасности (в том числе параметров поиска, доступа к разделам Решения). Настройка правил интеграции через подсистему интеграции (API). Создание новых и корректировка существующих объектов Решения посредством проектирования предметной области. Создание новых и настройка существующих схем регламентированных процессов
Оператор	Обладает ограниченным набором прав, включающим создание, просмотр и редактирование объектов Решения в разделах, разрешенных администратором (в соответствии с утвержденной ролевой политикой)	Просмотр информации об объектах. Создание новых экземпляров объектов. Редактирование информации об объектах
Авторизованный пользователь	Обладает правами просмотра, создания контента в разделах, разрешенных администратором.	Создание и просмотр информации об объектах.

4.1.3 Требования к показателям назначения

Время отклика при запросе к Решения не должно превышать 5 секунд.

Система должна функционировать 24 часа в сутки 7 дней в неделю за исключением технологических перерывов на обслуживание. Решение должно обеспечивать

одновременную работу не менее 50 000 пользователей. Должна быть предусмотрена возможность масштабирования Решения при увеличении нагрузки.

4.1.4 Требования безопасности

4.1.4.1 Требования к условиям окружающей среды

Требования к условиям окружающей среды не предъявляются.

4.1.4.2 Требования к кондиционированию

Для обеспечения необходимых климатических условий серверные помещения, в которых располагаются технические средства Решения, должны иметь систему кондиционирования.

Используемые системы кондиционирования должны обеспечивать возможность оповещения администраторов системы при невозможности поддержания заданных параметров окружающей среды.

4.1.4.3 Требования по электропитанию и заземлению

4.1.4.3.1 Для защиты от кратковременных перебоев электропитания все средства должны подключаться через источники бесперебойного питания достаточной мощности.

4.1.4.3.2 Защитное заземление должно обеспечивать защиту людей от поражения электрическим током при прикосновении к металлическим нетоковедущим частям, которые доступны для прикосновения человека и могут оказаться под напряжением в результате повреждения изоляции.

Защитное заземление должно выполняться преднамеренным электрическим соединением металлических частей электроустановок с «землей» или ее эквивалентом.

Заземляющие устройства должны выполняться по нормам на напряжение прикосновения или по нормам на сопротивление.

4.1.4.4 Требования к противопожарной безопасности

Требования к противопожарной безопасности не предъявляются.

4.1.5 Требования к эргономике и технической эстетике

4.1.5.1 Дизайн и оформление страниц Решения должны предусматривать использование следующих элементов оформления:

- четкие шрифты;
- общепринятые элементы навигации и иконки;
- графические объекты (рисунки, фото).

4.1.5.2 Навигационные средства Решения должны соответствовать следующим требованиям:

- пользователям должна предоставляться наглядная информация о структуре Решения;
- пункты меню должны называться или изображаться так, чтобы пользователь однозначно понимал их назначение;
- размер шрифта, используемый в интерфейсах, количество элементов на странице, их цвета, размеры и расположение не должно вызывать дискомфорт у пользователей Решения.

4.1.5.3 Взаимодействие пользователей Решения с прикладным программным обеспечением должно осуществляться посредством визуального графического интерфейса.

Ввод-вывод данных Решения, прием управляющих команд и отображение результатов их исполнения должны выполняться в интерактивном режиме.

4.1.5.4 Интерфейс не должен быть перегружен графическими элементами. Навигационные и управляющие элементы интерфейса должны быть выполнены в удобной для пользователя форме с соблюдением следующих условий:

- однозначность интерпретации наименований (наименование элемента должно позволять однозначно определить его назначение);
- унификация наименований (однотипные элементы должны иметь одинаковые наименования);
- унификация обозначений (однотипные элементы должны иметь одинаковые обозначения – графические значки, вид элементов управления);
- унификация использования (однотипные элементы должны иметь одинаковую реакцию на действия пользователя – наведение указателя, переключение фокуса, активации элемента управления).

Интерфейс должен быть рассчитан на преимущественное использование манипулятора типа «мышь», то есть управление должно осуществляться с помощью набора экранных меню, кнопок, значков и т.п. элементов.

Клавиатурный режим ввода должен использоваться главным образом при заполнении и/или редактировании текстовых и числовых полей экранных форм.

Все надписи экранных форм, а также сообщения, выдаваемые пользователю (кроме системных сообщений), должны быть на русском языке.

4.1.5.5 Решение должно обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях Решение должно выдавать пользователю соответствующие сообщения, после чего возвращаться в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

4.1.5.6 Экранные формы Решения должны проектироваться с учетом требований унификации:

- для обозначения сходных операций должны использоваться сходные графические значки, кнопки и другие управляющие (навигационные) элементы. Термины, используемые для обозначения типовых операций, а также последовательности действий пользователя при их выполнении, должны быть унифицированы;
- внешнее поведение сходных элементов интерфейса должны реализовываться одинаково для однотипных элементов.

4.1.5.7 Должна быть обеспечена возможность масштабирования экранных форм.

4.1.5.8 Интуитивно понятный пользовательский интерфейс должен обеспечивать минимизацию усилий пользователей при работе, что позволит:

- сократить длительность операций чтения и поиска информации;
- уменьшить время навигации и выбора команды;
- увеличить длительность устойчивой работы пользователей.

4.1.5.9 Цветовое решение интерфейса должно быть выдержано в спокойных тонах, не вызывающих утомление зрения.

4.1.5.10 Разработанный дизайн экранных форм должен соответствовать современным направлениям и тенденциям в веб-дизайне, а именно:

- адаптивность – адаптивный дизайн, который обеспечивает правильное отображение системы на различных устройствах, и динамически подстраивающийся под заданные размеры окна;
- простота – интуитивный, минималистический интерфейс;
- свободное пространство – принцип свободного пространства помогает подчеркнуть контент;
- плоский дизайн – принцип плоского дизайна представляет собой четкие визуальные решения, объекты изображаются схематично без передачи объёма;
- блоки – блочное расположение контента позволяет его адаптировать под размер окна браузера;
- большие изображения – использование больших изображений является современным трендом;
- большие кнопки – для удобства навигации используются крупные элементы (кнопки).

Для выполнения ввода типизированных значений по возможности должны использоваться компоненты, предназначенные для ввода значений указанного типа, с помощью которых пользователь может быстро вводить данные. Решение должно поддерживать ввод данных как и вручную, так и с помощью данных компонентов.

4.1.6 Требования к показателям надежности

4.1.6.1 В нормальном режиме Решения должна обеспечить круглосуточную и непрерывную работу при условии проведения требуемых технических мероприятий по обслуживанию Решения. При возникновении сбоя в программном обеспечении Решения после восстановления данные остаются консистентными.

4.1.6.2 Решение должно обеспечивать корректную обработку аварийных ситуаций, в том числе вызванных неверными действиями пользователей. Пользовательский интерфейс Решения должен отображать соответствующие сообщения об ошибках для пользователя, после чего Решение должно вернуться в рабочее состояние, предшествовавшее некорректному вводу данных.

4.1.6.3 Решение должно автоматически восстановить свою функциональность, в том числе в случае некорректного перезапуска программных и аппаратных средств.

4.1.6.4 При сбоях в системе электроснабжения технических средств Решения, приводящих к перезагрузке операционной системы, восстановление программных средств должно происходить автоматически после перезапуска операционной системы и запуска исполняемого файла Решения.

В программном обеспечении Решения должно быть предусмотрено:

- контроль целостности данных на уровне СУБД;
- сохранение целостности данных в базе данных при нештатном завершении работы Решения;
- автоматизированное восстановление Решения при корректном перезапуске технических средств;
- сохранение работоспособности программного обеспечения при некорректных действиях пользователя (ввод неверных по размеру и/или составу данных, нештатное завершение действий, выход из Решения до завершения действия);

- резервное копирование базы данных;
- журналирование операций Решения

4.1.6.5 Показатели надежности должны определяться прогнозируемой частотой возникновения аварийных ситуаций. Для системы регламентируются показатели надежности для следующих видов аварийных ситуаций:

- общесистемный отказ – выражается в недоступности всех или большинства пользовательских интерфейсов системы вне зависимости от причин, вызвавших этот отказ (отказы средств технического обеспечения, телекоммуникационных средств, общесистемного программного обеспечения, неверная работа специализированных программ, ошибки персонала, сбой электропитания и т.п.), кроме причин катастрофического характера (форс-мажорных обстоятельств);

- частный сбой – выражается в недоступности одного из компонентов, который вызывает недоступность одного или нескольких интерфейсов какого-либо компонента системы, или его некорректной работе. Время восстановления системы: до четырех часов, вызванных ошибками ПО, без учета восстановления дампов из резервных копий.

4.1.6.6 Требования по интенсивности отказов и длительности перерыва действительны при условии надлежащей численности и квалификации обслуживающего персонала, соблюдении ими технологических инструкций, а также при использовании установленных видов технического обеспечения, общесистемного и специализированного программного обеспечения.

4.1.6.7 Требования к надежности Решения уточняются по согласованию с заказчиком определенного проекта, реализуемого на базе Решения на стадии технического проектирования. При этом технические решения для удовлетворения указанных требований должны быть обоснованы на стадии разработки документации по проекту и отражены в соответствующей документации.

4.1.6.8 Требования к надежности технических средств и программного обеспечения

Надежность технических средств Решения должна обеспечиваться за счет резервирования оборудования.

Показатели надежности должны достигаться комплексом организационно-технических мер, обеспечивающих доступность ресурсов, их управляемость и обслуживаемость.

Технические меры по обеспечению надежности должны предусматривать:

- использование средств безопасности вычислительной сети и защиты информации;
- конфигурирование используемых средств с применением программного обеспечения, обеспечивающего высокую доступность;
- строгую регламентацию доступа к функциям подсистем.

Необходимо предусмотреть возможность обеспечения следующих технических мер повышения надежности:

- резервирование критически важных компонентов и данных и отсутствие возможности единовременного отказа дублирующих компонентов;
- организацию многоуровневой системы дублирования и архивирования информации;

- использование технических средств с избыточными компонентами и возможностью их «горячей» замены;

- использование системы комплексного администрирования.

Организационные меры по обеспечению надежности должны быть направлены на минимизацию ошибок персонала службы эксплуатации за счет:

- достаточной квалификации пользователей и обслуживающего персонала;
- регламентации и нормативного обеспечения выполнения работ персоналом;
- своевременного оповещения пользователей и обслуживающего персонала о случаях внештатной работы подсистем;

- минимизации ошибок персонала службы эксплуатации и при проведении работ по обслуживанию комплекса технических средств подсистем;

- ведения журнала отказов;

- минимизации времени ремонта или замены вышедших из строя компонентов за счет:

- регламентации проведения работ и процедур по обслуживанию и восстановлению подсистем;

- своевременной диагностики неисправностей;

- обслуживания и поддержки компонентов комплекса технических средств.

4.1.7 Требования по стандартизации и унификации

4.1.7.1 Должно быть обеспечено соответствие Решения требованиям следующих стандартов:

- HTTP 1.1 для обмена данными между уровнем бизнес-логики и презентационным уровнем;

- HTTPS для защищенного обмена данными между уровнем бизнес-логики и презентационным уровнем (при необходимости, в том числе в административном интерфейсе);

- ANSI SQL/92 для обмена данными между уровнем бизнес-логики и уровнем данных.

Унификация программных средств должна быть обеспечена за счет применения унифицированных компонент и средств из состава:

- общего и базового программного обеспечения;

- модуля «защищенное хранилище»;

- модуля «интеграционная шина».

4.1.7.2 При создании Решения следует руководствоваться действующими в Российской Федерации национальными стандартами и другими нормативно-техническими документами.

Проводимые работы должны соответствовать следующим требованиям:

- применяемые при создании Решения технические (форматы данных, протоколы передачи и прочие) и организационные (регламенты, требования, инструкции и т.п.) решения должны быть доступны и документированы в виде, достаточном для независимой реализации третьими сторонами. Применение недокументированных или недоступных решений не допускается;

- при выборе применяемых решений преимущество должно отдаваться решениям, основанным на стандартизированных технологиях, т.е. прошедшим процедуру

стандартизации и утвержденным в качестве стандарта либо рекомендации каким-либо признанным международным, федеральным, отраслевым, промышленным органом по стандартизации;

- экранные формы пользовательского интерфейса должны проектироваться с учетом требований унификации.

4.1.8 Требования к защите информации

Доступ пользователей к персонализированным функциям и данным Решения должен предоставляться только после прохождения пользователем процедур аутентификации и авторизации.

Доступ пользователей к функциям и данным Решения должен быть ограничен на основе ролевого принципа. Каждому пользователю Решения должна быть сопоставлена учетная запись, ассоциированная с одной из нескольких предопределенных пользовательских ролей. Для каждой пользовательской роли должны быть определены конкретные ограничения на доступ к функциям и данным Решения. Доступ к функциям чтения, записи, удаления, создания и навигации по объектам должен настраиваться исключительно пользователем, имеющим соответствующие права.

В целях обеспечения защиты Решения от несанкционированного доступа к административному интерфейсу и предотвращения размещения несоответствующей его назначению информации должна быть обеспечена возможность защищенного электронного обмена данными.

Параметры Решения должны проверяться на соответствие типам, размерам, значениям. Проверка параметров должна осуществляться до их передачи серверным модулям. Проверка параметров на соответствие не должна осуществляться только на клиентской стороне. Проверка должна осуществляться на разрешение только допустимых значений, а не по принципу исключения запрещенных значений. При доступе к закрытым областям проверка должна осуществляться при любой попытке доступа к любому адресу закрытой области. Недопустимо возникновение ситуации, при которой существует возможность получить доступ к объекту закрытой области, минуя проверку.

Конфигурация программного обеспечения должна исключать возможность несанкционированного доступа к Решения или отказа в обслуживании при максимальном использовании встроенных механизмов защиты информации.

Программное обеспечение не должно допускать возможности использования его для косвенного доступа к закрытым областям или данным. Не должно быть возможности получения содержимого закрытого раздела путем вызова открытых функций Решения с указанием адреса закрытого источника.

Комплекс мер по обеспечению сохранности информации и ее восстановления должен включать в себя:

- проведение регулярного регламентного копирования программного обеспечения и баз данных Решения в целом и ее подсистем;
- восстановление данных в непротиворечивое состояние при программно-аппаратных сбоях (отключение электрического питания, сбоях операционной системы и других) вычислительно-операционной среды функционирования;
- восстановление данных в непротиворечивое состояние при сбоях в работе сетевого программного и аппаратного обеспечения;

- проведение внепланового резервного копирования программного обеспечения и баз данных;
- хранение резервных копий на съемных носителях в специально оборудованных помещениях, обеспечивающих условия, исключающие их несанкционированное получение, хищение, подмену или уничтожение;
- восстановление программного обеспечения и баз данных из резервных копий.

4.2 Требования к формированию детализированных функциональных требований к разработке Решения

Должны быть формализованы требования к используемому в компонентах Решения математическому аппарату. По итогам данных работ должна быть составлена Алгоритмическая документация, содержащая описание используемых в Решении математических методов.

Должна быть разработана презентация концепции реализации Проекта.

4.3 Функциональные требования к компонентам (модулям) Решения

4.3.1 Функциональные требования к компоненту (модулю) электронного документооборота

Компонент электронного документооборота должен обеспечивать возможность организации формирования и согласования юридически значимого обмена документацией.

Компонент электронного документооборота должен состоять из следующих подсистем:

- Подсистема создания и доработки проектов документов;
- Подсистема согласования и подписания проектов документов;
- Подсистема регистрации и классификации документов;
- Подсистема управления и контроля.

4.3.1.1 Требования к функциям подсистемы создания и доработки проектов документов

Подсистема должна предоставлять возможность создания проектов документов и шаблонов документов и обеспечивать формирование и редактирование различных документов.

4.3.1.1.1 Функция создания и редактирования шаблонов документов

Подсистема должна реализовывать следующие функции для работы с шаблонами печатных форм документов:

- просмотр списка записей;
- фильтрацию записей;
- сортировку записей по полям;
- настройку отображения таблицы данных;
- выгрузку содержимого таблицы в файл Excel и PDF;
- просмотр записи;
- добавление записи;
- удаление записи;
- редактирование записи.

Подсистема должна обеспечить возможность создания и редактирования шаблонов печатных форм документов.

Должна быть возможность задания следующих характеристик:

- название шаблона;
- тип документа должен выбираться из списка.

Должна быть возможность загрузки файла шаблона и выбора типа шаблона (шаблон печати, шаблон письма).

4.3.1.1.2 Функция формирования проекта документа на основании шаблона (или без шаблона)

Функция формирования документов должна быть доступна пользователям при помощи интерфейса Системы.

Подсистема должна предоставлять возможность создавать проекты документов как на основе унифицированных (типовых) шаблонов документов для часто повторяющихся технологических или управленческих процессов, так и без использования шаблонов документов.

В результате формирования проекта документа подсистема должна присваивать проекту документа учетный номер, сформированный по заданному шаблону для выбранного типа/подтипа документа.

4.3.1.1.3 Функция редактирования/доработки проекта документа

При создании и доработке проекта документа подсистема должна предоставлять возможность прикрепления одного или нескольких файлов документов, созданных по шаблону.

При создании и доработке проекта документа подсистема должна предоставлять возможность прикрепления в качестве файла проекта документа сканированный образ документа, подготовленного на бумажном носителе.

Также подсистема должна предоставлять возможность просмотра любой версии любого прикрепленного файла.

Подсистема должна предоставлять возможность закрытия файла без сохранения внесенных изменений в файл проекта документа с одновременным снятием блокировки редактируемого файла.

Подсистема должна обеспечивать возможность пользователю, создавшему проект документа, удаления ранее прикрепленных файлов.

Должна быть обеспечена возможность связывания проекта документа с другим документом Системы с указанием типа связи. В этом случае Система должна формировать двустороннюю связь между документами: от связанного документа к связываемому и от связываемого документа к связанному.

Должна быть реализована возможность запрета завершения операции создания и доработки в том случае, если для документа не заполнены обязательные атрибуты.

При создании проекта документа подсистема также должна предоставлять возможность сохранения сформированного проекта документа в личной папке пользователя без отправки его по процессу подготовки. В этом случае подсистема также должна предоставлять возможность продолжения работы с сохраненным проектом документа и отправки его по процессу подготовки при необходимости.

4.3.1.1.4 Функция завершения подготовки документа

При создании и доработке проекта документа подсистема должна предоставлять возможность редактирования списка согласующих и подписантов. Также должна быть

реализована возможность аннулирования проекта документа и прекращения процесса его подготовки.

После завершения этапов создания и доработки подсистема должна обеспечивать отправку проекта документа на этапы согласования и подписания, в соответствии с заданной для проекта документа последовательностью этапов. Должна быть предоставлена возможность просмотра визуальной отчетности о стадиях согласования с указанием ответственных, сроков выполнения задач при работе с документом с цветовой индикацией просрочек действий пользователей относительно плановых сроков согласования.

При доработке проекта документа подсистема должна предоставлять возможность отправки документа как на повторное согласование, так и на продолжение согласования. При отправке на повторное согласование с этапа доработки документа подсистема должна направить задачу на согласование первой в списке очередей согласования. При отправке на продолжение согласования с этапа доработки документа подсистема должна направить повторно задачу на согласование на этап, с которого документ был отклонен.

Требования по отправке проекта документа на этапы согласования и подписания, реализации бизнес-логики взаимодействия процессов подготовки и доработки и процессов согласования и подписания должны быть реализованы средствами подсистемы управления регламентированными процессами.

4.3.1.2 Требования к функциям подсистемы согласования и подписания проектов документов

Подсистема должна предоставлять возможность согласования как в параллельном, так и в последовательном режимах и подписания проектов документов. При согласовании проекта документа должна быть возможность указать этапы и порядок согласования, в том числе перечень обязательных согласующих подразделений/ролей.

Подсистема должна предоставлять возможность добавления файлов, а также комментариев к файлам или замечаний к документу, к регистрационной карточке документа на этапах согласования и подписания.

Подсистема должна запрещать или давать возможность редактирования проекта документа на этапах согласования и подписания в зависимости от полномочий пользователя, назначаемых администратором Системы.

4.3.1.2.1 Функция настройки этапов и очередей согласования

Должна быть обеспечена возможность формирования этапов согласования и подписания как в шаблонах проектов документов на этапе создания, так и в проектах документов всех типов на этапах создания и доработки проекта документа.

Этапы согласования и подписания должны задаваться в виде последовательности очередей согласования и подписания с назначением на каждую очередь исполнителей (одного или нескольких) из числа пользователей Системы.

Должна быть обеспечена возможность формирования более одной очереди согласования и не более одной очереди подписания, как для шаблона проекта документа, так и для проекта документа на этапе создания проекта документа.

Назначение исполнителей на очередь согласования или подписания должно осуществляться посредством выбора пользователя из справочника организационно-штатной структуры.

При формировании этапов согласования и подписания для шаблона документа Подсистема должна обеспечивать создание шаблонов следующих видов:

- шаблон документа с заданной последовательностью очередей согласования и/или подписания и заданным списком согласующих и/или подписантов;
- шаблон документа с заданной последовательностью очередей согласования и/или подписания и пустым списком согласующих и/или подписантов;
- пустой шаблон документа, для которого последовательность очередей согласования и/или подписания не задана.

При формировании этапов согласования и подписания для шаблона документа или проекта документа на этапе создания проекта документа подсистема должна выполнять проверку корректности заданной последовательности очередей: очередь подписания не может располагаться до очереди согласования.

При создании проекта документа подсистема должна предоставлять возможность добавления очередей согласования и/или подписания, а также редактирования и удаления очередей, сформированных при создании проекта документа, и должна запрещать редактирование и удаление очередей согласования и подписания, наследуемых из шаблона документа.

При создании проекта документа подсистема должна обеспечивать обязательное указание одной очереди подписания.

Подсистема должна предоставлять возможность создания и редактирования очередей согласования и подписания на следующих этапах жизненного цикла документа:

- создание проекта документа;
- доработка проекта документа;
- согласование документа.

4.3.1.2.2 Функция управления прохождением документа по маршруту

Подсистема должна обеспечивать последовательное прохождение проекта документа по очередям согласования и подписания в порядке, заданном для проекта документа.

Должна быть реализована возможность указать этапы и порядок согласования, в том числе перечень обязательных согласующих

4.3.1.2.3 Функция управления действиями пользователей

При переходе проекта документа на этап согласования подсистема должна разослать одновременно всем исполнителям текущей очереди согласования задачу «На согласование».

При переходе проекта документа на этап подписания подсистема должна разослать задачи «На подписание» последовательно (т.е. по мере завершения задачи предыдущим исполнителем) исполнителям данной очереди в соответствии с порядком их следования в очереди подписания.

Подсистема должна запрещать или предоставлять возможность редактирования проекта документа на этапах согласования и подписания в зависимости от полномочий пользователя, назначаемых администратором. Требование должно быть реализовано с помощью функций подсистемы администрирования.

Подсистема должна предоставлять возможность добавления файлов, а также комментариев к файлам, к регистрационной карточке документа на этапах согласования и подписания.

При работе с проектом документа на этапах согласования и подписания подсистема должна предоставлять следующие возможности:

- согласовать/подписать проект документа;
- отправить проект документа на дополнительное согласование;
- отклонить проект документа и вернуть его на доработку с указанием причины отклонения.

Подсистема должна поддерживать возможность отправки проекта документа на дополнительное согласование, при котором согласующий делегирует рассмотрение проекта документа своим подчиненным.

Возможность отправки на дополнительное согласование должна предоставляться в виде специального компонента – «Дополнительное согласование», в котором указываются исполнители, которым направляется документ на дополнительное согласование. При этом подсистема должна автоматически сформировать в компоненте согласования новую очередь согласования.

Очередь должна быть дочерней по отношению к той, из которой инициировано дополнительное согласование. После отправки на дополнительное согласование текущая задача, из которой оно было инициировано, должна быть прекращена у пользователя. У пользователей, указанных при инициировании дополнительного согласования, должны появиться задачи «На дополнительное согласование».

В результате выполнения задач «На дополнительное согласование», пользователи либо согласуют документ, либо отклоняют его, а также вносят свои комментарии или прикрепляют необходимые файлы к задаче. Проект документа, после дополнительного согласования, должен быть возвращен на этап согласования для принятия решения согласующим (появляется задача «На согласование»).

При формировании этапов согласования и подписания для шаблона или проекта документа подсистема должна предоставлять возможность выполнения следующих операций над очередями этапов:

- создание новой очереди;
- удаление очереди;
- изменение последовательности выполнения очередей относительно друг друга;
- изменение состава исполнителей очереди (добавление в очередь исполнителей);
- удаление из очереди исполнителей;
- изменение последовательности выполнения очереди «Подписание» (исполнителями внутри очереди);
- редактирование очереди (редактирование наименования, изменение срока выполнения очереди).

При создании очереди согласования/подписания подсистема должна предоставлять возможность указания срока выполнения очереди в днях или часах с учетом календаря рабочего времени.

Подсистема должна запрещать/разрешать удаление очередей согласования / подписания на всех этапах жизненного цикла документа в зависимости от полномочий пользователя.

Требования к отзыву проекта документа:

- подсистема должна предоставлять возможность отзыва проекта документа с этапа согласования. Возможность отзыва должна быть предоставлена исполнителю проекта документа;

- при выполнении операции отзыва проекта документа с этапа согласования подсистема должна требовать ввод причины отзыва. Причина отзыва должна быть указана в ЭРК документа с указанием даты отзыва и причины отзыва документа;

- в результате выполнения операции отзыва проекта документа с этапа согласования, подсистема должна прекратить все незавершенные задачи по этапам у исполнителей задач;

- в результате выполнения операции отзыва проекта документа с этапа согласования подсистема должна формировать в листе визирования у всех согласующих, у которых были прекращены задачи при отзыве, соответствующий комментарий к визе согласования «Проект документа отозван исполнителем»;

- в результате выполнения операции отзыва проекта документа с этапа согласования подсистема должна направить проект документа на этап доработки;

- после доработки отозванного документа подсистема должна повторно направить документ всем согласующим на согласование.

В части требования к возврату проекта документа:

- подсистема должна предоставлять возможность возврата проекта документа на доработку с этапа согласования или подписания. Возможность возврата должна быть предоставлена визирующему или подписывающему проект документа;

- при выполнении операции возврата проекта документа с этапа согласования или подписания подсистема должна требовать ввод причины возврата. Причина возврата должна быть указана в ЭРК документа с указанием даты возврата и причины возврата документа;

- в результате выполнения операции возврата проекта документа с этапа согласования подсистема должна формировать в листе визирования у всех согласующих, которые возвращали документ на доработку, соответствующий комментарий к визе согласования «Проект документа возвращен согласующим»;

- в результате выполнения операции возврата проекта документа с этапа согласования или подписания подсистема должна направить проект документа на этап доработки;

- должны быть определены критерии доработки документа, позволяющие разделить на «существенные» и «несущественные» (например, ошибки, опечатки);

- после доработки возвращенного документа подсистема должна повторно направить документ всем согласующим на согласование в случае, если критерии определены как «существенные»;

- должна быть возможность гибкой настройки маршрута, участников, порядка направления документа на первоначальное и на повторное согласование, доступные администратору Системы.

4.3.1.2.4 Функция управления задачами

В рамках регламентированного процесса согласования и подписания проекта документа должна быть произведена настройка создания уведомлений.

При переходе проекта документа на этап согласования подсистема должна рассылать одновременно всем исполнителям текущей очереди согласования уведомление «На согласование».

При переходе проекта документа на этап подписания подсистема должна рассылать уведомление «На подписание» последовательно (т.е. по мере завершения подписания предыдущим исполнителем) исполнителям данной очереди в соответствии с порядком их следования в очереди подписания.

После отправки на дополнительное согласование у пользователей, указанных при инициировании дополнительного согласования, должны появляться уведомления «На дополнительное согласование».

Проект документа после дополнительного согласования должен возвращаться на этап согласования для принятия решения согласующим (появляется уведомление «На согласование»).

4.3.1.2.5 Функция управления визированием и подписанием документа

В результате выполнения этапов согласования и подписания проекта документа подсистема должна формировать лист визирования, в котором автоматически должен фиксироваться результат прохождения документом этапов согласования и подписания. При согласовании или подписании проекта документа в листе визирования должна отображаться виза «Согласовано» или «Подписано» соответственно, при отклонении проекта документа должна отображаться виза «Отклонено».

Подсистема должна предоставлять возможность вывода на печать листа визирования документа. В печатной форме листа визирования должна выводиться информация о списке согласующих/подписантов, выполнивших этап согласования / подписания, визу согласующего / подписанта, а также комментарий согласующего / подписанта. Вывод на печать листа визирования должен осуществляться с ЭРК документа.

При подписании (утверждении) подсистема должна позволять использовать квалифицированную усиленную ЭП и вносить изображение штампа ЭП в текст подписываемого (утверждаемого) документа.

4.3.1.3 Требования к функциям подсистемы регистрации и классификации документов

4.3.1.3.1 Функция классификации документов

Подсистема должна позволять выполнять регистрацию и классификацию документов, (входящих, исходящих, внутренних документов).

Основные сведения о входящем документе, которые должны быть отображены в ЭРК:

- наименование организации (корреспондента);
- наименование вида документа;
- дата документа / обращения;
- регистрационный номер документа / обращения;
- фамилия и инициалы лица, подписавшего документ;
- дата поступления документа / обращения;
- входящий / исходящий регистрационный номер;
- способ доставки документа;
- заголовок к тексту (краткое содержание документа / обращения);

- количество листов основного документа;
- отметка о приложении (количество приложений, общее количество листов приложений);
- сведения о связанных документах (наименование вида документа, дата, регистрационный номер, тип связи);
- резолюция (исполнитель / исполнители, поручение, должностное лицо, давшее поручение, дата формирования поручения);
- срок исполнения документа;
- индекс дела по номенклатуре дел;
- сведения о переадресации документа;
- отметка о контроле;
- гриф ограничения доступа к документу;
- сведения об электронной подписи;
- электронный адрес корреспондента;
- почтовый адрес (для обращений);
- вопрос по тематическому классификатору (для обращений);
- ссылки на файл поступившего документа и файл (файлы) приложения (приложений) к документу;
- и иные данные.

4.3.1.3.2 Функция связывания документов один с другим

При регистрации документов подсистема должна предоставлять возможность связывания документа с другим документом Системы с указанием типа связи. В этом случае Системы должна формировать двустороннюю связь между документами: от связанного документа к связываемому, и от связываемого документа к связанному.

4.3.1.3.3 Функция присвоения документу регистрационного номера и даты регистрации

После окончания процесса регистрации документа подсистема должна обеспечивать автоматическое присвоение документу регистрационного номера и даты регистрации.

Датой регистрации документа должна устанавливаться дата выполнения регистрации для документа (текущая дата).

Регистрационный номер для всех типов документов должен формироваться на основании заданного для типа/подтипа шаблона регистрационного номера.

Подсистема должна предоставлять возможность автоматического формирования регистрационного номера документа с присвоением очередного порядкового номера в течение года.

При регистрации документа подсистема должна предоставлять возможность прикрепления в качестве электронного образа документа одного или нескольких ранее созданных файлов, в том числе полученных путем сканирования.

Классификация документов должна осуществляться в соответствии с настроенным справочником видов и типов документов.

4.3.1.4 Требования к функциям подсистемы управления и контроля

Документы после их регистрации и рассмотрения руководством передаются в соответствующие структурные подразделения на исполнение.

Структурные подразделения получают доступ к документу и организуют работу с документом. При получении документа на исполнение, принятии к исполнению и в процессе работы с документом подсистема должна формировать уведомления ответственным лицам (взаимосвязь с подсистемой уведомлений) о ходе исполнения.

Контролю подлежат все зарегистрированные документы, требующие исполнения.

Контроль сроков исполнения документов (поручений) должен включать в себя:

- постановку документов (поручений) на контроль;
- проверку своевременности доведения документов (поручений) до исполнителей;
- предварительную проверку и регулирование хода исполнения документов (поручений);
- снятие с контроля документов (поручений);
- учет, обобщение и анализ результатов хода исполнения документов (поручений);
- информирование руководителей о ходе исполнения документов (поручений) и состоянии исполнительской дисциплины.

Срок исполнения документа должен указываться непосредственно в ЭРК документа. Напоминания исполнителям (уведомления), а также информация об исполнении документов (поручений), полученная от исполнителей, должны фиксироваться в ЭРК. Необходимо предусмотреть возможность получения напоминаний по электронной почте.

Подсистема должна выполнять контроль сроков согласования и исполнения поручений для следующих типов документов:

- входящие документы;
- исходящие документы;
- организационно-распорядительные документы (приказы и распоряжения);
- информационно-справочная документация (служебные, докладные и объяснительные записки, информационные письма).

Подсистема должна позволять изменять контрольные сроки поручений и контрольные сроки этапов процесса деятельности на основании запросов исполнителей поручений/этапа процесса, а также на основе входящей корреспонденции (например, ходатайства заявителя). Должна быть возможность гибкого изменения контрольных сроков посредством настроек Системы, доступ к которым предоставляется администратору Системы.

В случае нарушения контрольных сроков подсистема должна направлять соответствующие уведомления участникам процесса. Информация о нарушении контрольного срока также должна попадать в соответствующие отчеты по контролю исполнительской дисциплины.

Подсистема должна обеспечить контроль сроков согласования и исполнения поручений для документов, используя основные системные объекты и справочники.

Основные системные объекты подсистемы:

- Проект резолюции,
- Поручение,
- Отчёт по поручению,
- Отзыв поручения.

Справочник подсистемы:

- Типовые поручения.

Модель данных основных объектов подсистемы может быть уточнена на этапе разработки по согласованию с заказчиком.

Основные функции подсистемы должны включать в себя:

- подготовку проекта резолюции;
- утверждение резолюции;
- постановку документов на контроль;
- создание поручений;
- исполнение поручений.

Подсистема должна позволять пользователю создавать проект резолюции по документу, определять параметры резолюции и ответственных (заполнять соответствующие атрибуты), создавать произвольное количество поручений по документу в рамках резолюции, направлять резолюцию на утверждение руководителю (либо утверждать самому). Объект Проект резолюции должен ссылаться на документ, по которому создается резолюция. Переход на карточку проекта резолюции должен быть доступен из ЭРК документа.

Утверждение проекта резолюции должен производиться нажатием кнопки «Утвердить» в карточке проекта резолюции.

Постановка документа на контроль и снятие с контроля должно осуществляться непосредственно в ЭРК документа, где и отображается отметка о контроле.

Подсистема должна позволять создавать поручения по документу в рамках проекта резолюции (в таком случае поручения должны ссылаться на проект резолюции) или дочерние поручения в рамках родительского поручения (должны ссылаться на родительское поручение).

В поручении необходимо предусмотреть возможность прикрепления отчёта об исполнении поручения.

4.3.2 Функциональные требования к компоненту нормативно-справочной информации

4.3.2.1 При реализации проекта должна быть полностью сохранена функциональность компонента НСИ альфа-версии Решения.

4.3.2.2 Компонент должен выполнять следующие функции:

- создание группы справочников (классификаторов), в том числе линейных, иерархических, составных;
- изменение группы справочников (классификаторов);
- удаление группы справочников (классификаторов). При удалении группы все входящие справочники (классификаторы) должны переводиться в общую группу;
- обмен данными между справочниками и другими подсистемами (карточки, документы, редакции документов, прочая служебная информация);
- управление справочниками (классификаторами) с возможностью добавления/удаления атрибутов;
- сортировка объектов в справочнике по названию и реквизитам (атрибутам);
- поиск объектов в справочнике по названию и реквизитам (атрибутам);

– добавление в справочники и классификаторы данных из нормативно-правовой информации, требований стандартов, технических условий.

4.3.2.3 Конструктор (программный модуль) справочников и классификаторов должен позволять добавлять новые записи (типы и виды объектов), создавать многоуровневые иерархические структуры реестров, справочников и классификаторов.

4.3.2.4 Должна быть реализована возможность удалять записи по одной или по несколько, выбрав их с помощью соответствующей команды.

4.3.2.5 При просмотре удаленных записей должна быть возможность восстановить конкретную запись.

4.3.2.6 Функция поиска должна позволять выполнять поиск как по отдельным ключевым словам (быстрый поиск), так и составлять сложные запросы с логическими связями (расширенный поиск).

В режиме быстрого поиска пользователь должен иметь возможность задать искомую комбинацию символов. По умолчанию поиск введенной комбинации символов должен осуществляться по всем материалам, открытым для поиска пользователю с учетом связанности информации.

В режиме расширенного поиска система должна позволять пользователю сохранить настроенное логическое выражение в шаблон с целью последующего повторного использования. Расширенный поиск должен иметь возможность включения поиска по любым атрибутам объектов в любой их комбинации, указываемой в виде логических выражений с применением таких операторов объединения как «И» и «ИЛИ», а также скобок, указывающих очередность и приоритетность применения операторов.

В отношении шаблона поиска должна быть реализована возможность осуществлять следующие операции:

- сохранить;
- изменить;
- удалить;
- применить.

Должна быть реализована возможность осуществлять каскадный поиск (поиск в результатах поиска).

4.3.2.7 При отображении данных справочников должны быть обеспечены следующие возможности:

- сортировка по любому из атрибутов в справочнике;
- установка сортировки по умолчанию;
- фильтр по любому из атрибутов в справочнике;
- выбор из перечня преднастроенных и сохранение собственных фильтров;
- возможность выгружать данные справочника (частично и/или полностью) в форматах xls,xlsx;
- индивидуальная настройка параметров отображения для каждого пользователя.

4.3.2.8 Требуется провести интеграцию компонента в архитектуру Решения.

За счет использования микросервисной архитектуры доработка компонента при интеграции в архитектуру Решения не требуется.

4.3.3 Функциональные требования к интеграционной шине

4.3.3.1 Решение должно включать предустановленные интеграции.

Интеграционная шина поддерживает три основных механизма интеграции:

- через обмен файлами (xlsx, csv и т.д.);
- через общую СУБД;
- через обмен сообщениями (REST, SOAP, GraphQL).

Интеграционная шина включает конструктор метаданных, позволяющий настраивать структуру данных, получаемых из внешних источников. Конструктор позволяет обрабатывать, как простейшие скалярные значения одного типа, так и сложные структуры, включающие несколько типов данных.

4.3.3.2 Должна быть обеспечена поддержка современных инструментов настройки информационного взаимодействия с внешними системами и аппаратно-программными комплексами для обеспечения Решения возможности корректного встраивания в текущую ИТ-инфраструктуру потенциальных Заказчиков, а также для организации корректного информационного обмена с внешними источниками данных для формирования целостной информационной картины.

Разработаны специальные адаптеры для интеграции с:

- ЕСИА;
- СМЭВ;
- Active Directory;
- информационные системы, поддерживающими SOAP/REST;
- почтовыми сервисами POP3, Secure POP3, IMAP, Secure IMAP, SMTP, Secure SMTP);
- IP-телефонией, SiP, E1, WebRTC;
- механизмами шифрования и усиленной квалифицированной электронной подписи.

4.3.3.3 Должны быть обеспечена возможность интеграции компонента интеграционной шины Решения с распространенными BI-решениями иностранных вендоров с целью обеспечения корректной миграции данных при импортозамещении продуктов на российском рынке.

Ниже представлен список BI-решений иностранных вендоров, с которыми должна быть обеспечена интеграция:

- 1) Power BI;
- 2) Qlik Sense.

Интеграция возможна в рамках конкретных проектов, реализуемых на базе Решения с учетом требований заказчика, только при условии предоставлении со стороны заказчика реквизитов доступа к API, с входными параметрами для API методов по получению источников данных.

4.3.3.3 Требования к функциям (задачам) подсистемы интеграции (API)

Модуль интеграции должен иметь следующую функциональность:

- ведения базы метаданных, включающей описания структуры единой базы данных, источников информации, мест хранения информации, протоколов информационного обмена, форматов представления документов;
- загрузка как структурированных, так и неструктурированных данных с их последующим размещением в подсистеме хранения данных;

- загрузка информации из источников в виде текстовых, табличных файлов или медиа-файлов (изображения, видео, аудио);

- возможность работы со всеми основными типами источников информации (HTTP веб-сервисы, базы данных, xml, плоские файлы, потоковые данные с различных датчиков и устройств, логи поведения пользователей и проч.), а также со всеми основными форматами данных;

- поддержка возможности организации защищенных каналов передачи данных.

Модуль должен обеспечивать выполнение следующих функций:

- обеспечение возможности поддержки синхронного и асинхронного способа вызова служб;

- обеспечение возможности использования защищенного транспорта;

- обеспечение возможности доступа к данным для внешних информационных систем (пользователей системы) с помощью специально разработанных адаптеров;

- обеспечение возможности обработки и преобразования сообщений;

- обеспечение контроля процессов информационного обмена (аудиты, протоколирование).

Для обеспечения юридически значимого документооборота должна быть реализована возможность использования электронной подписи.

Модуль должен обеспечивать:

- автоматический асинхронный (по запросу) и синхронный (по расписанию) сбор, предобработку и классификацию информации от заданных администратором источников (совместно с подсистемой интеграции);

- инкрементальные методы загрузки информации;

- минимизацию объема обрабатываемой информации за счет удаления дублей;

- автоматическую классификацию и фильтрацию данных по таким признакам,

как:

- дата получения информации;

- объект, с которого была получена информация;

- группам пользователей, которым доступна информация;

- другим признакам (должны настраиваться администратором или ответственным лицом);

- получение оценок по передаваемым данным.

Модуль должен позволять:

- выполнять настройку параметров опроса источников/приемников данных;

- выполнять настройку схемы метаданных, загружаемых из источника или передаваемых в приемник данных;

- автоматически формировать метаданные на основе XML/JSON файлов;

- в соответствии с настроенными схемами проверять целостность и корректность данных;

- выполнять автоматическое наполнение информационных объектов системы, в случае настроенного отображения («мэппинга») полей, загружаемых данных на поля внутренних объектов;

- журналировать процессы по загрузке и передаче данных.

Модуль должен включать следующие функциональные блоки:

- блок планировщика для управления расписаниями запуска процедур опроса источников. Данный модуль должен выполнять расчеты времени запуска опроса для каждого источника и осуществлять запуск загрузчиков для каждого источника;
- блок выбора источника для взаимодействия с планировщиком, определения заданных источников, времени их запуска;
- блок загрузки данных для осуществления базовой навигации и выполнения загрузки данных через подсистему интеграции. Должен быть разработан конструктор правил импорта и преобразования данных для каждого источника;
- блок преобразования данных к единому виду для структурирования информации и добавления к ней метаданных;
- блок обнаружения сбоев для выявления случаев некорректной работы загрузчика в связи с изменением структуры источника или API внешних автоматизированных систем. Должна быть реализована двухуровневая проверка корректности загружаемых информационных материалов;

Модуль должен обеспечивать доступ к Системе через протоколы REST и SOAP для обмена данными с внешними информационными платформами.

4.3.4 Функциональные требования к компоненту (модулю) преобразования информации ETL

4.3.4.1 Наличие компонента преобразования информации (ETL) должно обеспечивать гибкие возможности по извлечению данных из внешних источников, их первичной проверки и оценки возможности последующей загрузки в хранилище данных Решения, распределению данных на потоки с преобразованием данных к виду, соответствующему эталонной модели данных Решения.

4.3.4.2 ETL-сервис должен выполнять весь цикл загрузки, выгрузки и преобразования данных через веб-интерфейс.

С помощью ETL-сервис должны выполняться извлечение, преобразования и загрузку данных силами администратора/оператора Системы. Управление всем циклом обработки данных происходит в графическом конструкторе регламентированных процессов (Workflow).

ETL сервис должен проводить автоматическое выполнение всех типовых задач ETL:

- протоколирование выполнения задачи ETL;
 - настройка очередности выполнения задачи ETL;
 - извлечение данных из двумерных, реляционных и многомерных источников данных;
 - извлечение/загрузка данных из внешних источников: OLE DB; баз данных Microsoft Access, DBase; файлов Microsoft Excel, XML, HTML, TXT, CSV, JSON;
 - извлечение/загрузка данных из объектов репозитория: таблиц, присоединенных таблиц, представлений, запросов, источников данных ODBC, журналов;
 - извлечение данных из REST-источников, доступ к которым осуществляется через Internet;
 - преобразование данных: слияние, разделение, удаление дубликатов, группировка, фильтрация, сортировка.
 - выгрузка данных в необходимый формат для передачи во внешнюю систему.
- ETL сервис может быть использован для:

- управления информационным обменом с внешними системами;
- обработки и преобразования плоских и иерархичных данных для получения и передачи через API;

- настройки разовой миграции большого количества данных.

4.3.4.3 Встроенный конструктор должен позволять мониторинг и контроль процесса исполнения, оповещая о неправильном использовании ресурсов, порядке и статусе обработки данных.

ETL сервис должен обеспечивать извлечение больших данных из различных источников, их дальнейшее прототипирование, анализ, проверку, преобразование и загрузку в любое новое место назначения.

Подраздел ETL-сервиса предоставляет набор готовых сценариев и мастеров и состоит из следующих пунктов меню:

- Реестр для выгрузки;
- Реестр загрузок;
- Xslt редактор (редактор/преобразователь файлов форматов xsl, xml);
- Реестр преобразованных данных (регистры ETL);
- ETL статусы.

4.3.4.4 В ходе развития ETL-компонента должен быть доработан интерфейс XSLT-схем. В верхней части карточки (горизонтально) должны располагаться основные команды для работы с формой («Восстановить», «Пресет», «Импорт/экспорт пресета», «Отчеты») и команда для сохранения вносимой информации – «Сохранить».

Форма (карточка) должна содержать поля:

- «Заголовок» – для ввода наименования адаптера загрузки (схемы преобразования);
- «Активация функции» – активация адаптера загрузки в реестре схем;
- «Ввод Xslt» – загрузка схемы преобразования из файлового хранилища или другого репозитория.

В верхнем правом углу должны располагаться кнопки команд «Сохранить» и «Сохранить и закрыть».

Необходимо модернизировать интерфейс конструктора формирования XSLT-схем посредством добавления следующих методов:

- NewGuid – генерация уникального GUID в рамках текущей трансформации;
- Trim – удаление пустых символов с начала и конца строки;
- Replace – замена подстроки во входной строке;
- ClearEscapeChars – экранирование спец символов (") во входной строке;
- GetArrayObjects – метод для загрузки связанных с исходным объектом ManyToManyAssociation;
- GetEasyCollection – метод для загрузки связанных с исходным объектом;
- GetLocation – метод для загрузки типа данных Location;
- DateTimeAddOffset – загрузка объекта DateTime со заданным смещением в часах;
- GetDaysByDateTime - метод для загрузки типа данных Date;

4.3.4.5 Должны быть разработаны обучающие материалы, обеспечивающие методологическую поддержку пользователей компонента.

Обучающие материалы должны содержать информацию о создании нового ETL-процесса. А именно:

Шаги формирования ETL-процесса на загрузку:

- подготовка загружаемых данных;
- создание ETL-процесса (бизнес-процесса);
- загрузка данных (запуск бизнес-процесса).

Если загружаемые данные в формате XML, то подготовки не требуется, если в других форматах, например, json/excel-таблица, то требуется преобразование в XML.

Процесс подготовки данных для Excel-таблиц:

- описать структуру данных в виде XSD-схемы;
- импортировать XSD-схему в Excel;
- сопоставить схему со столбцами;
- экспортировать данные.

Создание ETL-процесса:

- сформировать XSL-схему трансляции данных;
- сформировать БП для загрузки.

Формирование XSL-схемы:

- открыть пункт меню «Конструктор схем сопоставления»;
- загрузить подготовленный XML-файл с исходными данными (не более 1 Мб);
- сформировать схему сопоставления данных.

Формирование бизнес-процесса:

- перейти в модуль бизнес-процессов;
- создать БП с указанием типа объекта «FileEtlProcess» – сущность, отвечающая за загрузку данных из XML в систему;
- в редакторе БП создать схему из двух блоков:
- DownloadCommand – выполняет операцию загрузки;
- DownloadMessage – выполняет операцию ожидания загрузки.

Шаги формирования ETL-процесса на выгрузку данных из системы:

- создание ETL-процесса;
- выгрузка данных.

Создание ETL-процесса:

- сформировать БП для выгрузки;
- сформировать XSL-схему выгрузки данных;
- выгрузить данные из системы.

4.3.5 Функциональные требования к компоненту (модулю) автоматизированного проектирования предметной области

4.3.5.1 Компонент автоматизированного проектирования предметной области построен на базе конструктора объектного проектирования (low-code), который поддерживает возможность кодогенерации классов, что обеспечивает высокую степень адаптивности программного решения.

Конструктор low-code позволяет создавать новые классы, на основе которых создаются объекты – экземпляры классов (экземпляры объектов заданного вида), из административного интерфейса без привлечения разработчиков.

Конструктор автоматизированного проектирования предметной области должен поддерживать следующий функционал:

- создания/модификации/удаления класса объекта;
- добавления/модификации свойств (полей) к классу;
- управления правами доступа свойств (полей) класса;
- настройки правил создания и уничтожения экземпляра класса.

Механизм кодогенерации должен обеспечивать следующие функции:

- ввод параметров нового класса (представления модели данных):
 - основных данных:
 - имени;
 - наименования;
 - параметров сборки;
 - параметров использования в регламентированных процесса
 - параметров доступа
 - параметров отображения экземпляров класса
 - полей:
 - имени;
 - заголовка на экранных формах;
 - заголовка и порядкового номера в списке;
 - вкладки, на которой осуществляется вывод;
 - ограничений (максимальная длина);
 - параметров доступа (обязательное, только для чтения, скрытое);
 - параметров связей;
- трансляция параметров модели в машинный формат;
- формирование табличного представления объекта в базе данных;
- сборка связей, внешних и внутренних ссылок в базе данных;
- настройка прав доступа и схем движения объектов класса.

Конструктор проектирования предметной области должен обеспечивать добавление в модель класса полей следующих типов:

- int;
- bool;
- date;
- datetime;
- decimal;
- double;
- string;
- text;
- file;
- image;
- enum;
- агрегация;
- ссылка;
- коллекция ассоциаций.

Конструктор конфигураций (проектирования) предметной области должен позволять создавать и отображать новые объекты предметной области (классы, подмножества) наиболее оптимальным для работы пользователя методом.

Конструктор конфигураций (проектирования) предметной области должен обеспечивать:

- унификацию пользовательских интерфейсов;
- специализацию пользовательских интерфейсов под требуемые задачи;
- удаление из интерфейса служебной информации, которая ухудшает восприятие сущности;
- исключение параметров сущности, замедляющих работу пользователей;
- наследование ограничений, понятий и связей одной конфигурации у другой с помощью автоматизации операции наследования;
- создание перегруженных процедур обработки и вывода данных конфигурации.

Процесс создания новых конфигураций состоит из следующих шагов:

- выбор базовой конфигурации;
- задание имени новой конфигурации;
- настройка формы отображения конфигурации;
- настройка отображения списка данных;
- настройка процедур фильтрации отображаемых данных;
- настройка наименования конфигурации, формы и отображения;
- настройка доступа к конфигурации в системе.

4.3.5.2 В рамках реализации проекта должна быть проведена интеграция компонента автоматизированного проектирования предметной области в общую архитектуру Решения.

4.3.5.3 Должна быть выполнена разработка функциональности программного компонента в части упрощения интерфейса.

Конструктор автоматизированного проектирования предметной области должен обеспечивать гибкие возможности по оперативному созданию новых сущностей (объектов) программных решений без использования программирования.

Должна быть добавлена настройка параметров типа добавляемого свойства в объект, параметры визуального отображения карточки экземпляра объекта.

Должна быть добавлена настройка карточки объекта, для динамического изменения структуры карточки и упрощения интерфейса.

4.3.5.4 Должен быть разработан комплекс обучающих материалов для обеспечения методической поддержки пользователей, начинающих эксплуатацию программного продукта, включающий в себя следующую информацию:

- типы данных;
- типы связей;
- создание новой модели
- параметры объекта:
 - создание объекта;
 - свойства объекта;
 - создание атрибута;
 - создание зависимости объекта;

- ошибки публикации.

4.3.6 Функциональные требования к компоненту (модулю) защищенного распределенного хранения данных с встроенными редакторами документов

4.3.6.1 Компонент защищенного распределенного хранения данных с встроенными редакторами документов в Решения должен обеспечивать возможность безопасного распределенного хранения данных с возможностью ограничения доступа к файлам, их перемещения между дисками и папками, а также должен поддерживать контроль версионности материалов с журналированием событий в системе.

4.3.6.2 Компонент должен обладать единым интерфейсом доступа ко всем информационным объектам, обеспечивать эффективное распределение репозитория по кластерным узлам, определение доступных пользователю репозитория.

4.3.6.3 Компонент должен обеспечивать резервное копирование данных и сохранение их копий по физически распределенным кластерным узлам.

4.3.6.4 Компонент защищенного распределенного хранения данных с встроенными редакторами документов в Решения должен быть интегрирован в единую архитектуру Решения.

4.3.6.5 Компонент должен обеспечивать:

- хранение информации с учетом квотирования предельного размера для хранения, предельного размера загружаемого файла в виде файлов форматов *.rtf, *.doc, *.docx, *.txt, *.xls, *.xlsx, *.png, *.gif, *.jpg, *.jpeg, *.ico, *.bmp, *.avi, *.mp4, *.xml, *.pdf и редактируемой метаданных (тегов) к файлам для поиска и краткого описания;

- просмотр и редактирование файлов форматов *.rtf, *.txt, *.docx, *.xlsx встроенными средствами;

- просмотр и рецензирование файлов формата *.pdf встроенными средствами;

- обеспечение версионности документов с возможностью просмотра и восстановления предыдущих версий;

- хранение удаленных документов в «корзине»;

- шифрование хранимой информации в соответствии с требованиями ГОСТ Р 34.12-2015;

- настройку ограничений на максимальный размер диска/папки/загружаемого файла.

4.3.6.6 При работе с виртуальными дисками должны предусматриваться возможности по:

- созданию диска (стандартного или зашифрованного);

- определению предельного размера для диска и размещаемых на дисках файлов;

- загрузке папок и файлов на диск;

- переименованию диска;

- отображению свойств диска в отдельном окне с указанием информации по предельному размеру диска и использованному размеру (как сумма объема расположенных файлов);

- удалению диска;

- настройке отображения реестра дисков в виде дерева или в табличной форме с отображением сведений о диске: наименование, тип, объем, состояние и иные с возможностью фильтрации, сортировки и поиска.

4.3.6.7 Функции работы с папками должны предусматривать:

- создание папки;
- загрузку файлов и папок в папку с выводом сообщения об успешной или неуспешной загрузке, содержащего имя и размер папки или файла, при неуспешной загрузке – системные и иные ограничения в загрузке;
- архивирование содержимого папки (в формат Zip)/разархивирование;
- скачивание папки в архивированном виде (в формат Zip);
- перемещение папки в другую папку или на другой диск;
- копирование папки в выбранную папку или диск;
- переименование папки;
- отображение свойств папки в отдельном окне (размер (как сумма объема расположенных документов), расположение, владелец, дата и время создания и т.д.);
- добавление папки в «избранное»;
- удаление папки;
- настройка отображения нескольких папок на диске или другой папке в виде галереи отдельных иконок.

4.3.6.8 Функции работы с документами должны предусматривать:

- предварительный просмотр файла (если предварительный просмотр недоступен, должно выдаваться соответствующее сообщение и должна предоставляться возможность скачать файл);
- редактирование/рецензирование встроенными средствами с учетом формата файла;
- перемещение документа в другую папку или диск;
- переименование документа;
- отображение свойств документа в отдельном окне;
- копирование документа в выбранную папку или диск;
- скачивание файла;
- архивирование файла документа (в формат Zip)/разархивирование;
- удаление документа;
- добавление документа в «избранное».

4.3.6.9 Компонент защищенного распределенного хранения данных должен обеспечивать возможность добавления отдельных пользователей или нескольких пользователей в хранилище данных системы. При добавлении пользователя должна быть обеспечена возможность настройки разграничений доступа пользователя к дискам, папкам, файлам.

Компонент защищенного распределенного хранения данных должен обеспечивать возможность извлечения сложных взаимосвязанных комплектов электронных документов из системы с возможностью предварительного формирования архива данных с возможностью массовой выгрузки пакетов данных.

4.3.7 Функциональные требования к компоненту (модулю) управления регламентированными процессами

4.3.7.1 Компонент управления регламентированным процессами (BPM), в основе которого лежит WorkFlow-конструктор, обеспечивает гибкие возможности по оперативной настройке регламентированных процессов, в том числе процессов формирования,

согласования и утверждения отчетности и иных сопутствующих материалов. Наличие данного компонента в «Visary BI» обеспечивает значительную гибкость Решения.

Конструктор автоматизации регламентированных процессов (Workflow) реализует возможность конструирования регламентированных процессов без знания языков программирования с интуитивно понятным графическим интерфейсом.

Должно быть реализовано формирование и отображение карты движения в соответствии с настроенным регламентированным процессом и соответствующими комментариями ответственных лиц.

Основой Компонента согласования отчетности, определяющей базовый набор сущностей и атрибутов для описания процессов, является ее метамодель, которая должна включать:

- общее описание процессов и типов взаимосвязей;
- множество логических шагов (задач и подпроцессов) процессов, описанных в четырех измерениях: кто, что, когда и как;
- описание участников процессов, отражающее организационную модель их взаимодействия;
- навигацию (взаимодействие) между задачами и подпроцессами в процессах;
- описание приложений, представляющих собой отображения объекта на каждом шаге или этапе процесса в любой момент времени;
- релевантные (актуальные) данные, полученные в результате выполнения шага процесса, связанные с отображением и влияющие на выбор следующего шага.

4.3.7.2 В рамках мероприятий по развитию компонента управления регламентированными процессами (BPM) должна быть проведено обновление интерфейсов конструктора Workflow (рисунок 3).

Панель блоков управления бизнес-процессами должна иметь обновленную структуру, содержащую следующие разделы:

- Основное:
 - Инициализатор
 - Этап
 - Подпроцесс
 - Триггер
 - Ожидание события
 - Действия
 - Условие
 - Условие (контекст)
- Управление объектами:
 - Создание
 - Изменение
 - Удаление
 - Взаимодействие
 - Создание на основании
 - Параллельное выполнение
- Коммуникация с пользователем
- Внутренние действия

Панель инструментов должна быть оптимизирована и содержать следующие элементы:

- Импорт/экспорт схемы
- Печать
- Уменьшить масштаб
- Увеличить масштаб

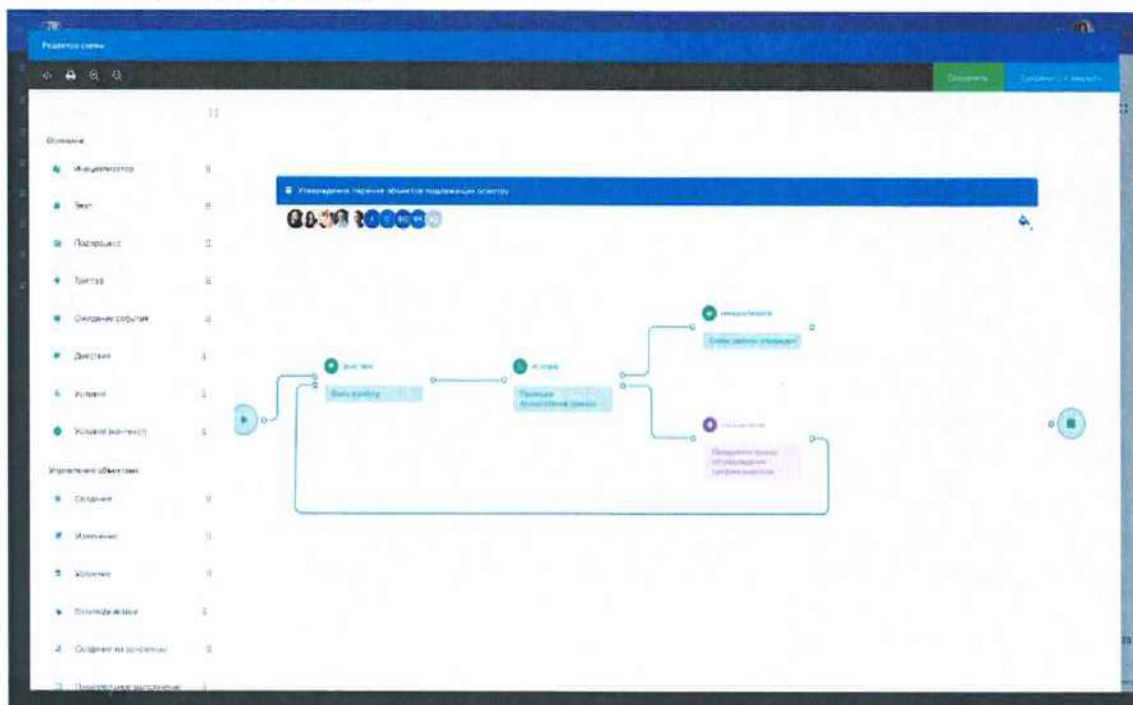


Рисунок 3 – Прототип интерфейса конструктора Workflow

Должна быть добавлена панель настроек, содержащая информацию о пользователях, группах пользователей (рисунок 4).

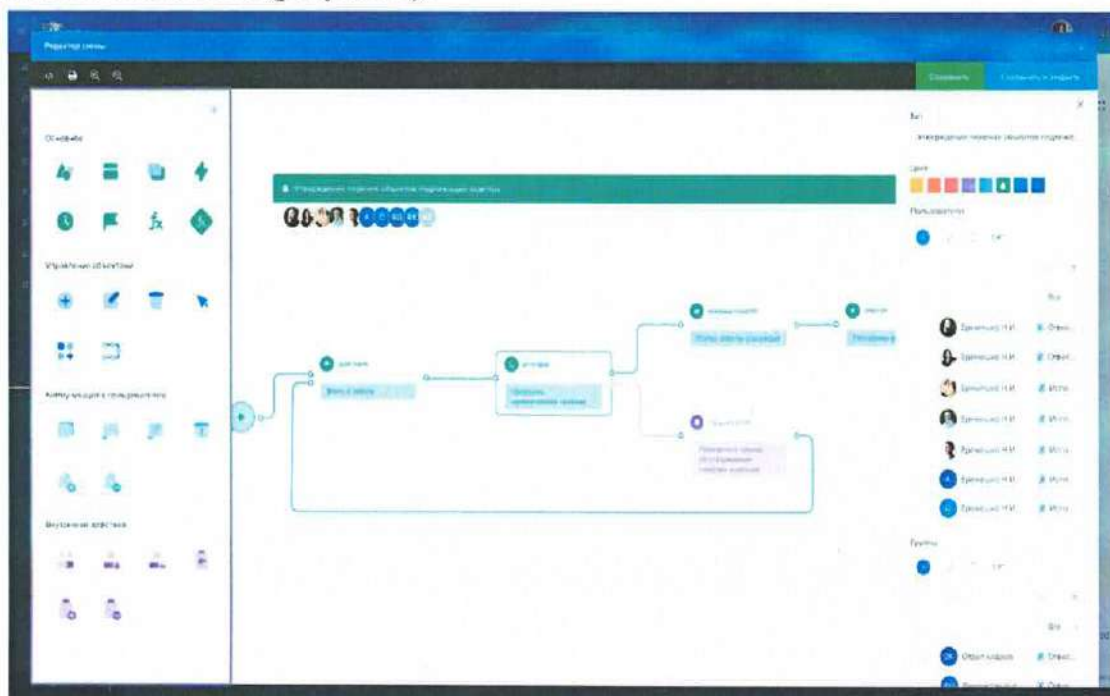


Рисунок 4 – Прототип интерфейса конструктора Workflow с панелью настроек, содержащую информацию о пользователях

4.3.7.3 Должен быть разработан интерфейс форм настройки согласования отчетности и иной сопутствующей документации для обеспечения возможности вывода на печать листов согласования.

Форма настройки согласования отчетности должна содержать следующую информацию:

- тип очереди
- название очереди
- тип согласования
- срок исполнения
- исполнитель
- длительность.

Должна быть предусмотрена возможность согласования непосредственно в листе согласования, где пользователь может при настройке доступа к объекту выбрать следующие кнопки:

- согласовать,
- отклонить,
- делегировать.

Должна быть обеспечена возможность пользователю оставлять комментарии (рисунок 5).

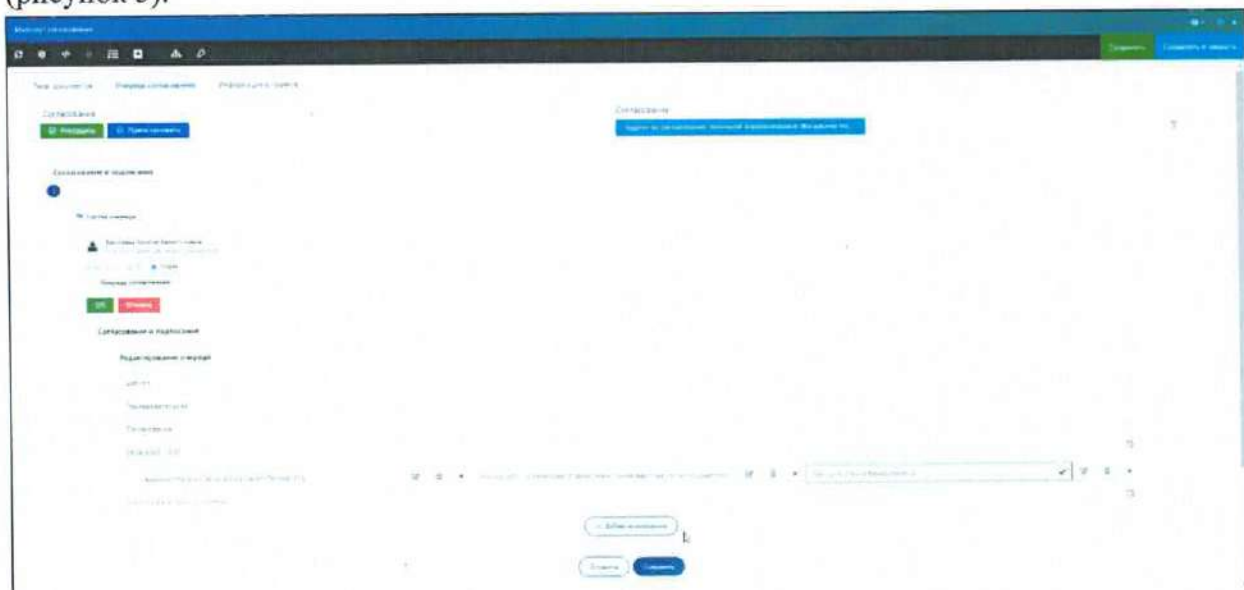


Рисунок 5 – Создание комментария

Должна быть реализована возможность создания шаблонов пользователем. Каждый пользователь может создать шаблон для себя, включая администратора и обычных пользователей. Выбор шаблона должен быть доступен в листе согласования с помощью функционального компонента «Выбор шаблона».

4.3.7.4 Должна быть проведена интеграция компонента с остальными структурными элементами Решения для формирования универсального решения многомерного анализа и визуализации больших данных «Визари СЭД» (Решения).

4.3.8 Функциональные требования к компоненту (модулю) управления задачами

4.3.8.1 В рамках реализации Проекта должна быть оптимизирована структура карточки задачи. Должен быть сформирован банк карточек задач в зависимости от категории задачи. Задачи должны быть подразделены на следующие категории:

- По статусу задачи:
 - новая
 - назначена
 - в работе
 - на тестировании
 - на согласовании
 - согласована
 - отклонена
 - решена.
- По исполнителю:
 - я исполнитель
 - мои задачи на исполнении.
- По автору.
- В зависимости от роли в команде:
 - куратор
 - наблюдатель
 - соисполнитель
 - соавтор.
- По приоритету:
 - высокий
 - средний
 - низкий.
- По типу задачи:
 - аналитика
 - дизайн
 - разработка
 - BI.

4.3.8.2 При проведении разработки компонента (модуля) управления задачами должны быть сохранены функциональные требования компонента по работе с массивами задач альфа-версии Решения.

4.3.8.3 Модуль должен являться основным инструментом взаимодействия руководителей и подчиненных. Процесс назначения, исполнения и контроля задач и поручений становится эффективным и прозрачным как для управленческого персонала предприятия, так и для исполнителей.

Модуль управления задачами должен обеспечивать автоматизацию процессов постановки задач с указанием ответственных и сроков исполнения, а также обеспечивать контроль их исполнения.

Модуль управления задачами должен автоматизировать:

- распределение задач (например, между сотрудниками Заказчика, подведомственных учреждений и подрядчиками);
- дифференциацию задач по приоритетам и времени исполнения;
- уведомление о критическом состоянии задачи;
- контроль выполнения задач (например, между сотрудниками Заказчика, подведомственных учреждений и подрядчиками).

Модуль управления задачами должен обеспечивать:

- создание задач в ручном режиме;
- автоматическое создание шаблонных задач при выполнении проектов или запуске регламентированных процессов только для пользователей, указанных в них исполнителями;

- разделение задач на категории (входящие, исходящие, выполненные);
- выбор приоритета задачи;
- отображения прогресса выполнения задачи в процентах;
- выбор исполнителя задач;
- отображение пользователя, назначившего (создавшего задачу);
- возможность прикрепления к задаче файлов любого типа;
- возможность связывания задачи с любым информационным объектом системы;
- ведение истории изменения задачи;
- напоминания о просроченных задачах;
- цветовое ранжирование задач по срочности (например, срочность подготовки проведения закупок/поставок).

Модуль управления задачами должен обеспечивать следующие возможности:

- создания задачи с привязкой к конкретному объекту;
- интеграция с компонентом управления регламентными процессами;
- интеграция с компонентом уведомлений. Исполнитель задачи получает уведомление о назначенных на него новых задачах;

- выбора ответственного исполнителя среди пользователей системы.

Предусмотрена возможность назначения задачи на группу пользователей (отдел). При этом пользователи, входящие в число исполнителей, обладают возможностью взять задачу на исполнение. В этом случае ответственный исполнитель изменяется на конкретного пользователя системы;

- построения отчетов по завершенным и открытым как входящим, так и исходящим задачам;

- выбора даты начала и плановой даты завершения задачи из календаря;
- автоматического расчета времени, оставшееся до завершения задачи. В случае просрочки рассчитывается время просрочки. Просроченные задачи визуальным образом отделены от остальных путем цветовой дифференциации в общем стеке задач исполнителя.

- Возможность выбора приоритета задачи для отделения наиболее важных и срочных задач. Возможные приоритеты:

- высокий;
- нормальный;
- низкий.

- Возможность цветовой дифференциации любым цветом из матрицы RGBA.

- Возможность ведения статусов задач. В базовом варианте предусмотрены следующие статусы:

- Новая;
- В работе;
- Отклонена;
- Закрыта;
- Решена;
- Назначена.

Список статусов может быть дополнен:

- предусмотрена возможность ведения истории задачи с указанием исполнителей, дат совершения действий над задачей, а также комментария пользователя;
- возможность прикрепления файлов любых форматов к задаче.

В компоненте управления задачами должен быть реализован объектно-ориентированный конструктор схем согласования задач. Конструктор позволяет настраивать:

- наименование схемы движения задач;
- подробное описание схемы;
- статус задачи для каждого этапа;
- действие, выполняемое на каждом этапе;
- параметры перехода:
 - начальный этап;
 - конечный этап;
 - прогресс задачи;
 - необходимость ввода комментария;
- параметры уведомления:
 - роль пользователя, для которого создается уведомление;
 - текст уведомления;
- параметры отображения полей задачи.

Конструктор должен включать функционал по созданию и редактированию схем согласования задач в графическом (визуальном) режиме.

В Компоненте должна быть реализована функция динамического распределения нагрузки с учетом текущей занятости и требований официальных регламентов. Распределение нагрузки позволяет распределять работы между исполнителями.

Компонент управления задачами должен интегрироваться с компонентом управления регламентированными процессами и компонентом уведомлений для автоматизации назначения и контроля действий, требуемых от исполнителя, и выполнять следующие функции:

- создание новой задачи в указанной группе;
- просмотр и изменение параметров существующей задачи;
- выбор ответственного исполнителя, соисполнителей и наблюдателей из числа пользователей Системы;
- задание плановой даты начала и завершения задачи;
- автоматический расчет времени, оставшегося до завершения задачи, с уведомлением о просрочке;
- выбор приоритета задачи (высокий, низкий, нормальный);
- отображение всех задач пользователя в общем стеке задач, с возможностью сортировки и фильтрации по простому и составному ключам;
- определение статуса задачи;
- ведение истории задачи (сохранение даты действия, исполнителя и комментариев пользователя);
- прикрепление к задаче файлов любых форматов и текстового описания;
- сопоставление задаче категории новой или существующей;

- сопоставление текущей задаче родительской из числа предшествующих задач с установкой иерархической связи;
- изменение даты завершения задачи и смена исполнителя;
- создание новой схемы движения задачи или привязка уже существующей.

Визуальный объектно-ориентированный конструктор схем движения задач должен позволять:

- создавать схемы, посредством размещения графических примитивов на рабочем поле конструктора;
- назначать схеме уникальное имя и сохранять ее для повторного использования;
- открывать и редактировать ранее созданные схемы;
- сопоставлять статусы задачи для каждого этапа;
- создавать связи между этапами, ассоциированные с выполняемыми действиями по изменению статуса;
- определять параметры перехода (наименование действия, начальный и конечный этап, прогресс задачи);
- настраивать уведомления о совершении действий пользователем в соответствии с его ролью;
- настраивать параметры отображения полей задач;
- выполнять сортировку и фильтрацию действий по заданному критерию.

4.3.8.3 Должна быть проведена интеграция компонента в архитектуру Решения.

4.3.8.4 Должна быть разработана модель расчета текущего показателя выполнения задач на основании математических алгоритмов.

Компонент управления задачами предназначен для автоматизации процессов постановки задач и контроля их исполнения. Компонент управления задачами содержит конструктор схемы движения задач, который обеспечивает возможность четкого отображения процесса перемещения задачи между этапами и функциональных механизмов, влияющих на выбор траектории перемещения под воздействием внешних и внутренних факторов. Конструктор обеспечивает визуализацию зависимостей между задачами для чего необходимо обеспечить соответствие схемы следующей математической модели.

Схема движения S задачи представляет собой тройку $S = (M, \mu, A)$, где M – семейство модификаций иерархических сетей Петри, каждая из которых соответствует множеству траекторий развития задачи, $M = \{C_1, C_2, \dots, C_i, \dots, C_q\}$, $z \in [1, q]$ – индекс корневой сети, с которой начинается функционирование схемы, μ – начальная маркировка C_z , определяющая исходный статус задачи. Каждая сеть C_i представляет собой шестерку $C_i = (P, T, I, O, A, D)$, где $P = \{p_1, p_2, \dots, p_i, \dots, p_n\}$ – множество позиций сети C_i , $T = \{t_1, t_2, \dots, t_i, \dots, t_m\}$ – множество переходов, I – входная функция, O – выходная функция, $A = \{A_1, A_2, \dots, A_i, \dots, A_k\}$ – множество активационных функций сети C_i , эквивалентных действию по установке начального состояния задачи, $D = \{D_1, D_2, \dots, D_i, \dots, D_l\}$ – множество деактивационных функций, эквивалентных действию по завершению выполнения задачи.

Структурно сеть C_i представляет собой множество позиций, соединенных между собой переходами. Позиция характеризуется числом содержащихся в ней фишек, инкапсулирующих условие перевода задачи на следующий этап, а переходы выполняют функции перемещения фишек по сети. Посредством фишек регламентируется подмножество операций по изменению статуса задачи, доступных в текущей ситуации.

Переход может являться или атомарным действием (независимый переход), или сложной последовательностью действий, описываемых иерархически подчиненной сетью (подчиненный переход). Запуск перехода t_i возможен только в том случае, когда в проассоциированных с ним посредством входной функции I позициях (входные позиции) содержится достаточное количество фишек. Результатом успешного запуска перехода является увеличение на определенное число количества фишек в позициях, проассоциированных с переходом t_i посредством выходной функции O (выходные позиции). Выходная функция определена на двумерном множестве, являющемся декартовым произведением пространства переходов на пространство деактивационных функций иерархически подчиненной сети (при ее существовании). В случае неуспешного запуска перехода t_i , возможного только в ситуации использования иерархически подчиненной сети, увеличение количества фишек в соответствующих позициях не производится. При этом в зависимости от значения признака завершения, определяемого деактивационной функцией, количество фишек во входных позициях перехода t_i может оставаться прежним или уменьшаться. Позиции соответствуют статусам задачи, а переходы – работам, оказывающим влияние на прогресс задачи и изменяющим ее состояние. Активационная функция A_i сопоставляет сети C_i некоторую начальную маркировку μ , определяющую стартовое количество фишек в позициях множества P . Деактивационная функция D_i представляет собой пару $D_i = \langle \mu', r \rangle$, где μ' – некоторая маркировка, достижение которой является сигналом к завершению работы сети, а двойка $r = \langle s, f \rangle$ – определяет итог завершения работы сети (результат $s \in \{0, 1\}$, где 0 – неуспешное завершение, 1 – успешное завершение, а признак результата $f \in \{0, 1, 2\}$, где 0 – успешное завершение, 1 – неуспешное завершение без изменения количества фишек во входных позициях, 2 – неуспешное завершение с уменьшением количества фишек во входных позициях). Срабатывание деактивационной функции сети эквивалентно достижению одного из конечных состояний задачи, то есть ее выполнению (в случае подчиненной сети – выполнению этапа).

Таким образом, произвольный переход t_i представляет собой тройку $t_i = \langle v, C', A' \rangle$, где $v \in \{0, 1\}$ – тип перехода (0 – независимый, 1 – подчиненная сеть), C' – иерархически подчиненная сеть, используемая для определения успешности запуска перехода t_i , A' – активационная функция сети C' ($A' \in A$ из C'). $C' = \emptyset$ и $A' = \emptyset$ в случае $v = 0$. Пример рассматриваемой сети C_i представлен на рисунке 6.

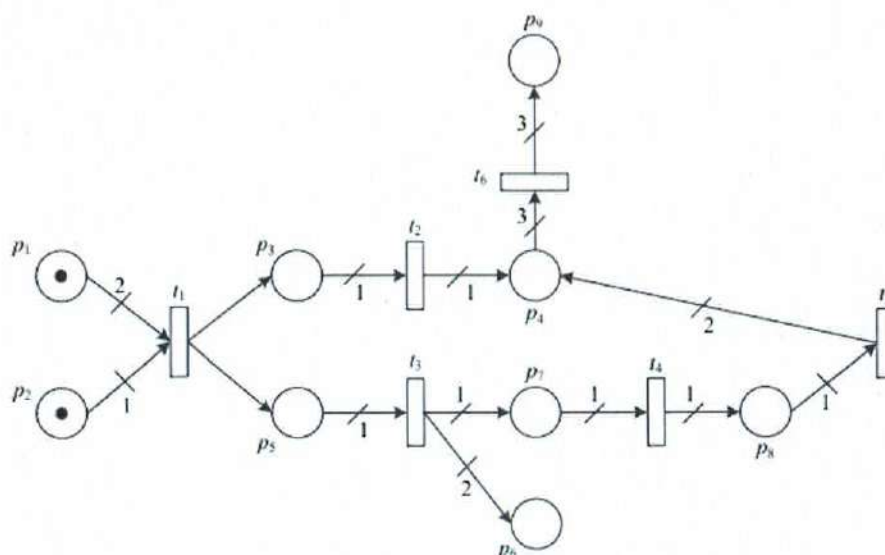


Рисунок 6 – Пример используемой модификации иерархической сети Источник: составлено авторами

Позиции $p_1, p_2, \dots, p_9$ соответствуют этапам движения задачи (этап 1, этап 2, ...). Переходы $t_1, t_2, \dots, t_6$ эквивалентны работам, оказывающим влияние на прогресс задачи и изменяющим ее статус. Переход t_1 является подчиненным и ему соответствует иерархически вложенная сеть Петри. Переходы $t_2 - t_6$ независимы. На исходящих дугах указано количество фишек, перемещающееся в выходные позиции в случае активации перехода. Для перехода t_1 количество фишек в выходных позициях будет определяться деактивационной функцией вложенной сети.

Модель расчета текущего показателя выполнения задачи должна быть расширена следующими сценариями:

1) Процент выполнения задачи должен вычисляться по следующей формуле:
 Процент выполнения задачи = $(100\% / (\text{Дата окончания} - \text{Дата начала} + 1)) * \text{Текущий день выполнения задачи (попадающий в период выполнения задачи)}$.

2) Для случаев, когда у родительской задачи (эпика) имеются дочерние подзадачи, у которых указаны трудозатраты на ее выполнение, то процент выполнения задачи рассчитывается по формуле:

Процент выполнения задачи = $(\text{Общий вес закрытых дочерних подзадач} / \text{Общий вес дочерних подзадач, входящих в состав родительской задачи}) * 100\%$.

4.3.8.5 После интеграции данного компонента с компонентом аналитической отчетности необходимо разработать банк предустановленных отчетных форм, наглядно отображающих степень эффективности работы различных сотрудников и подразделений в целом.

Банк предустановленных отчетных форм должен содержать следующие разделы:

- эффективность работы сотрудников
- эффективность работы подразделений
- статистика по проектным задачам.

4.3.9 Функциональные требования к компоненту (модулю) уведомлений

4.3.9.1 В рамках реализации Проекта должна быть проведена интеграция компонента уведомлений с другими элементами Решения.

Модуль уведомлений должен быть интегрирован с модулем бизнес-процессов (регламентированных процессов) и модулем расписаний (планировщиком задач).

Принцип работы модуля уведомлений основан на том, что при возникновении любого события, связанного с пользователем, такого как создание задачи, завершение согласования документа или его возврат на доработку и т.п., то есть при переходе бизнес-процесса от одного участника к другому у исполнителя автоматически формируется уведомление, представляющее графическое, текстовое и звуковое оповещение.

Компонент уведомлений обеспечивает автоматизацию следующих процессов:

- оповещение по электронной почте, в личных кабинетах пользователей системы о событиях, связанных с процессами согласования отчетности;
- рассылка сообщений иного тематического содержания.

При прочтении уведомления, а также при ознакомлении с произошедшим событием (например, при прочтении новой задачи), уведомление автоматически переносится в архив.

Компонент уведомлений обеспечивает рассылку уведомлений по электронной почте тем пользователям, которые выбрали для себя данную функцию. Для предотвращения перехвата информации, содержащейся в уведомлениях, реализована возможность использования e-mail через SSL.

4.3.9.2 Должна быть проведена разработка шаблона уведомлений, генерируемых на почту пользователям Решения в виде различных интерактивных отчетов с перечнем задач (поставленных, просроченных и от руководителя) с возможностью перехода в систему при нажатии на соответствующий раздел отчета.

Для рассылки сообщений (уведомлений) пользователям на электронную почту по результатам выполнения шагов схем бизнес-процессов (например, результаты процессов рассмотрения, согласования, голосования, утверждения и т.д.) или о других различных событиях используются настройки почтового сервера.

Для изменения настроек необходимо выбрать поле для изменения.

Для изменения полей с вводом данных необходимо выбрать поле для изменения и ввести желаемое значение.

Для изменения полей «Включено/выключено» необходимо установить переключатель в нужное положение.

4.3.10 Функциональные требования к компоненту (модулю) поиска

4.3.11.1 В Решения должен быть обеспечен полный перенос функционала компонента альфа-версии Решения в части переноса функциональности по поиску, сортировке и фильтрации массива данных Решения для формирования соответствующей выборки.

Компонент поиска должен позволять выполнять поиск как по отдельным ключевым словам (быстрый поиск), так и составлять сложные запросы с логическими связями (расширенный поиск).

В режиме быстрого поиска пользователь должен иметь возможность задать искомую комбинацию символов. По умолчанию поиск введенной комбинации символов осуществляется по всем материалам, открытым для поиска пользователю с учетом связанности информации.

В режиме расширенного поиска компонент должен позволять пользователю сохранить настроенное логическое выражение в шаблон с целью последующего повторного использования. Расширенный поиск должен обеспечивать возможность включения поиска

по любым атрибутам объектов в любой их комбинации, указываемой в виде логических выражений с применением таких операторов объединения как «И» и «ИЛИ», а также скобок, указывающих очередность и приоритетность применения операторов.

4.3.11.2 В отношении шаблона поиска должна быть реализована возможность осуществлять следующие операции: сохранить, изменить, удалить, применить.

4.3.11.3 Компонент поиска должен позволять осуществлять каскадный поиск (поиск в результатах поиска).

При отображении списков данных обеспечены следующие возможности:

- сортировка по любому из атрибутов в списке;
- установка сортировки по умолчанию;
- фильтр по любому из атрибутов в списке;
- выбор из перечня преднастроенных и сохранение собственных фильтров;
- возможность выгружать список (частично и/или полностью) из Компонента поиска в форматах xls,xlsx;
- индивидуальная настройка параметров отображения для каждого пользователя.

4.3.11.4 Компонент поиска должен позволять оперативно строить сложные запросы, по которым информация, представленная в системе в виде информационных объектов, отбирается в списки информационных объектов, отвечающих заданным критериям поиска. При этом в поиске должна учитываться вся организация модели данных бизнес-процессов и связанность информации в ней.

При открытии реестра объектов в виде таблицы должен быть предусмотрен поиск, фильтрация, сортировка и группировка по полям (колонкам) таблицы, возможность настройки порядка вывода полей и группировки.

В модуле (компоненте) поиска предусмотрена настройка перечня выводимых на экран полей таблицы для каждого пользователя (индивидуальный поиск).

Компонент поиска позволяет настраивать фильтры не только для отображаемых полей, но и для скрытых полей мнемоник, а также связанных объектов.

Разработать экранные формы для эффективной работы с фильтрами. На основной панели инструментов (позволяющей выполнять операции создания, удаления, редактирования записей) каждого окна, содержащего множества объектов, представленных в табличной или списочной формах, представлено поле выбора фильтра.

При создании фильтра пользователем задаются следующие параметры:

- наименование фильтра (краткое описание);
- описание фильтра (расширенное описание);
- критерии фильтрации данных.

Реализованы следующие функции:

- создание фильтров;
- редактирование фильтров;
- сортировка фильтров;
- экспорт фильтров в формат xlsx.

4.3.11.5 Должна быть проведена интеграция компонента поиска в общую архитектуру Решения.

4.3.11 Функциональные требования к компоненту (модулю) безопасности

Компонент безопасности (защиты информации) должен быть разработан с учетом требований стандарта ГОСТ Р 51583-2014 «Защита информации. Порядок создания автоматизированных систем в защищенном исполнении. Общие положения».

Компонент безопасности должен представлять единую точку входа Интернет / Интранет (извне и внутри корпоративной сети заказчика) для всех ответственных лиц заказчика, иных пользователей системы, имеющих отношение к процессам подготовки, согласования отчетности.

Компонент безопасности должен представлять инструменты управления учетными данными пользователей, которые позволят настраивать роли, объединять их в группы и разграничивать доступ к функциям для конкретной роли, пользователя и группы пользователей, а также осуществление распределения видимости объектов по ролям, пользователям, группам пользователей, группам ролей.

Компонент безопасности должен представлять механизмы:

- противодействия попыткам несанкционированного доступа к защищаемой информации;
- пресечения попыток несанкционированного считывания, изменения и уничтожения данных;
- контроля и восстановления целостности данных.

Компонент безопасности должен обеспечивать:

- возможность создания групп доступа и групп пользователей с заданными правами доступа, к которым относятся:
 - полный доступ к внутренним и внешним разделам альфа-версии/прототипе Решения;
 - право на создание и редактирование задач, справочников, документов;
 - право на удаление задач, справочников, документов;
 - право на просмотр задач, справочников, документов;
- возможность назначения прав доступа к задачам, справочникам в соответствии с иерархической организационной структурой функциональных заказчиков, включая определение прав доступа на основе принадлежности к определенному подразделению;
- разделение прав доступа по задачам, справочникам, папкам, объектам, по роли пользователя, по принадлежности к подразделению и занимаемой пользователем должности;
- возможность идентификации и проверки подлинности субъектов доступа при входе в Систему на основе альфа-версии/прототипе Решения (обеспечение доступа только зарегистрированных пользователей, прошедших процедуру идентификации и аутентификации);
- регистрацию (журналирование) входа субъектов доступа в Систему на базе альфа-версии/прототипа Решения (при регистрации указываются параметры: дата и время входа, идентификатор субъекта);
- доступ к функциям чтения, записи, удаления, создания объектов и навигации по объектам настраивается исключительно пользователем, имеющим соответствующие права;
- реализацию функции, позволяющей вести мониторинг работы пользователей, выполнять автоматический контроль нормального состояния функционирования системы, задействовать функции отслеживания и фиксации нештатных и аварийных ситуаций,

создавать механизмы, обеспечивающие возможность восстановления работоспособности после отказов, сбоев или аварий;

- внесение в журнал таких событий, как добавление, изменение, удаление данных пользователями;
- изменение записей журнала доступно только администратору;
- доступ к функции блокировки учетной записи администратора при управлении учетными записями администраторов без удаления из базы данных;
- интеграцию компонента (модуля) безопасности в архитектуру Решения для обеспечения комплексной защиты, включающей в себя защиту от сбоев, ведущих к потере информации, а также защиту от неавторизованного создания или уничтожения данных, обрабатываемой и хранимой в программных продуктах информации, в том числе персональных данных пользователей и иной конфиденциальной информации.

4.3.12 Функциональные требования к компоненту (модулю) администрирования

Компонент администрирования должен позволять администратору настраивать параметры функционирования ПО в целом или отдельных компонентов:

- настройка видимости разделов административной панели;
- настройка компонента безопасности: уровней логирования, аудита и журналирования;
- настройка требований к учетным записям;
- настройка интеграции с внешними системами.

Компонент администрирования должен обеспечивать удаленное журналирование операций, в том числе действий пользователей.

Компонент администрирования должен реализовать возможность анализа журнала на предмет выявления потенциальных нарушений доступа, фильтрации и сортировки журнала.

Компонент администрирования должен обеспечивать организацию доступа пользователей к информации в соответствии с правами доступа, а также реализовывать механизм разграничения доступа к функциям системы в зависимости от роли пользователя в системе. Должна быть реализована возможность создавать и редактировать пользовательские роли. Роль включает в себя перечень доступных пользователю функций и прав доступа к информации. При настройке (создании и редактировании) роли, реализована возможность указывать доступные отчеты, в том числе встроенные отчеты.

4.3.12.1 В ходе реализации проекта должна быть актуализирована рабочая и техническая документация с учетом нового функционала Решения.

4.3.12.2 Интерфейс Решения должен быть локализован для работы на английском языке.

4.3.12.3 На английском языке должны быть разработаны переведены следующие эксплуатационные и технические документы:

- архитектура (дизайн) программного обеспечения (software architecture design document);
- документация по API (API documentation);
- руководство конечного пользователя (end-user guide);
- руководство системного администратора (system administration documentation);
- инструкция по установке (installation instruction).

4.4 Требования к видам обеспечения Решения

4.4.1 Требования к математическому обеспечению

При создании математического обеспечения Решения Исполнитель должен использовать расчетные алгоритмы, которые обеспечат реализацию и работу всех реализуемых функций.

При описании расчетных алгоритмов должны быть указаны:

- назначение и характеристика алгоритма;
- математическое описание;
- алгоритм решения.

При аналитической обработке должны использоваться методы статистической обработки и анализа данных:

- фильтрации данных (по задаваемому пользователю набору и комбинациям фильтров);
- суммирования (в том числе по заданной временной и объектной выборке);
- определения среднего (арифметического, геометрического, средневзвешенного и т.п.);
- определения параметров статистических распределений, в том числе индексов динамики процессов и расчета трендов;
- расчета коэффициентов корреляции между различными показателями.

4.4.2 Требования к информационному обеспечению

4.4.2.1 Информационное обеспечение должно представлять собой совокупность сообщений, форм, документов, а также информационных баз данных (массивов информации), включая нормативно-справочную информацию. Состав информационного обеспечения должен быть полным и достаточным для решения задач заказчика.

Информационное обеспечение должно быть достаточным для поддержания всех автоматизируемых функций объекта автоматизации.

4.4.2.2 Информационное обеспечение должно соответствовать следующим требованиям:

- масштабируемость, без ограничения на количество подключаемых пользователей;
- структурированное хранение информации, с учетом поддержки версионности и истории изменения данных, с возможностью хранения истории изменения данных. Состав данных должен быть определен на этапе технического проектирования;
- непротиворечивость и целостность хранимой информации.

4.4.2.3 Состав, структура и способы организации данных должны быть детализированы для определенного проекта, реализуемого на базе Решения.

В состав информационного обеспечения Решения должны входить:

- данные, вносимые пользователями;
- данные, загружаемые из внешних информационных систем;
- данные, формируемые в объекте автоматизации в рамках автоматизируемых процессов;
- системные данные, в частности настройки, используемые компонентом.

4.4.2.4 С точки зрения доступа в Решения должно быть предусмотрено хранение, ведение и предоставление сведений ограниченного доступа и сведений, доступ к которым

не ограничен. Доступ к данным ограниченного доступа должен быть предоставлен только авторизованным пользователям с учетом их полномочий.

4.4.2.5 Для хранения данных в Решения должна быть использована открытая СУБД PostgreSQL.

4.4.2.6 Для индексации и кеширования должны быть использованы NoSQL СУБД Redis и поисковая база данных Elasticsearch.

4.4.2.7 Для обмена данными с внешними должны быть использованы следующие адаптеры:

- ЕСИА;
- СМЭВ;
- Active Directory;
- технология SOAP/REST.

4.4.3 Требования к программному обеспечению

Решение должно быть реализовано с использованием следующего технологического стека:

- ОС Linux или Astra Linux;
- СУБД PostgreSQL, Redis;
- сервер приложений ASP.NET Core;
- веб-сервер Nginx;
- экранные формы должны быть разработаны в соответствии с условиями предоставления пользователям полноценной работы в следующих браузерах для персональных компьютеров:

- Google Chrome версии 66.0 и выше;
- Mozilla Firefox версии 59.0 и выше;
- Opera 53.0 версии и выше;
- Яндекс.Браузер версии 10 и выше.

4.4.4 Требования к лингвистическому обеспечению

Должны применяться языки программирования, обеспечивающие работу в используемой СУБД, а также Unix-подобных ОС.

В качестве языка манипулирования данными должен быть использован язык структурированных запросов SQL (Structured Query Language). Выбор конкретного языка манипулирования данными зависит от используемой СУБД.

Для разработки веб-интерфейса должны быть использованы такие средства, как HTML5 и CSS.

Для разработки интерактивных веб-страниц должны быть использованы такие технологии, как ajax и json, средства языка JavaScript и библиотеки JQuery.

Интерфейс пользователя клиентского ПО должен быть на русском и английском языках.

Размещение информации в Решения должно осуществляться на русском и английском языках.

Основными языками взаимодействия Решения с обслуживающим персоналом (с учетом ограничений, налагаемых системным и базовым программным обеспечением) должны быть русский и английский языки.

Подсистемы Решения должны обеспечивать:

- интерфейс взаимодействия с пользователем на русском и английском языках;

- предоставление пользователю отчетов на русском и английском языках;
- формирование текстов сообщений об ошибках и мониторинговых сообщений (кроме сообщений системного и базового программного обеспечения) на русском и английском языках. Сообщения должны быть информативными и понятными пользователю.

При публикации документов должна обеспечиваться возможность чтения документов потребителями информации в кодировке UTF-8.

4.4.5 Требования к техническим средствам

Бета-версия Решения может быть установлена и запущена на физическом или виртуальном сервере, имеющем следующие характеристики:

- количество процессоров: 2;
- количество ядер процессора: 4;
- частота процессора: 1,3 ГГц;
- объем жесткого диска 2500 ГБ;
- объем оперативной памяти: 8 ГБ;
- скорость подключения 100 Мбит/с.

Для эффективной работы приложения на основе платформы «Визари» должен использоваться физический или виртуальный сервер, имеющий следующие характеристики:

- количество процессоров: 4;
- количество ядер процессора: 4;
- частота процессора: 2,4 ГГц;
- объем жесткого диска 21000 ГБ;
- объем SSD 2*256 ГБ;
- объем оперативной памяти: 32 ГБ;
- скорость подключения 1000 Мбит/с.

5 Состав и содержание работ по разработке Решения

5.1 Требования к детализированной модели монетизации Решения

В рамках детализированной модели монетизации Решения:

- изучена динамика и тенденции развития мирового и российского рынков EDMS;
- определен перечень зарубежных и российских EDMS-решений для анализа;
- выявлены критерии сравнения EDMS-решений на основе исследования мнения пользователей EDMS-систем;
- проведена сравнительная характеристика выбранных зарубежных и российских EDMS-решений по выявленным критериям;
- описана модель монетизации Решения с учетом результатов проведенного анализа зарубежных и российских EDMS-решений.

При проведении последовательного и глубокого анализа рынка существующих российских и иностранных EDMS-решений для выявления лучших практик реализации функциональных возможностей EDMS-решений для наиболее оптимального формирования детализированных требований к функциональности Решения.

5.2 Требования расчетным алгоритмам, используемым для Решения

При создании математического обеспечения Решения Исполнитель должен использовать расчетные алгоритмы, которые обеспечат реализацию и работу всех реализуемых функций.

При описании расчетных алгоритмов должны быть указаны:

- назначение и характеристика алгоритма;
- математическое описание;
- алгоритм решения.

Расчетные алгоритмы должны быть представлены в документе «Алгоритмическая документация, содержащая описание используемых в Решения математических методов».

5.3 Порядок разработки и этапы работ по разработке Решения

Этапы и сроки реализации Проекта уточнены в таблице 4.

Таблица 4 – Этапы и сроки реализации Проекта

№	Описание мероприятия	Результат (в т.ч. количественно измеримый)	Подтверждающие документы	Сроки реализации Проекта
Этап 1				
1.1	Мониторинг аналогичных российских и иностранных EDMS-решений. Анализ решений конкурентов с целью выявления лучших практик для совершенствования собственного Решения	Проведен последовательный анализ аналогичных существующих российских и иностранных EDMS-решений, определена наиболее востребованная со стороны пользователей функциональность (сильные стороны существующих программных продуктов).	Детализированное описание модели монетизации Решения по итогам оценки условий, предлагаемых разработчиками других программных продуктов по автоматизации документооборота, а также внутренних расчетов себестоимости разработки Решения	С 16.01.2023 по 31.01.2023
1.2	Формирование детализированных требований к разработке программного обеспечения Решения	Формализованы детализированные функциональные требования к разработке программного обеспечения Решения. Формализованы требования к используемому математическому аппарату. Разработана презентация концепции реализации Проекта.	– Детализированное техническое задание; – Алгоритмическая документация, содержащая описание используемых в Решения математических методов; – Презентация концепции реализации Проекта	
Этап 2				
2.1	Разработка компонентов Решения	Проведена разработка Решения согласно требованиям Детализированного технического задания	– Ведомость технического проекта; – Пояснительная записка к техническому проекту; – Программа и методика приемочных испытаний.	С 01.02.2023 по 21.09.2023

№	Описание мероприятия	Результат (в т.ч. количественно измеримый)	Подтверждающие документы	Сроки реализации Проекта
Этап 3				
3.1	Тестирование Решения	Проведено функциональное и нагрузочное тестирование Решения. Разработана пользовательская эксплуатационная документация.	– Руководство пользователя Решения; – Руководство администратора Решения; – Руководство для интегратора, содержащее рекомендации по развертыванию и интеграции Решения, организации мониторинга его работоспособности; – Программа и методика приемочных испытаний; – Акт о завершении опытной эксплуатации и готовности Решения к приемочным испытаниям; – Полный актуальный дистрибутив Решения, записанный на внешний носитель; – Протокол приемочных испытаний.	С 22.09.2023 по 25.09.2023
3.2	Регистрация программного обеспечения	Зарегистрированные в Роспатенте свидетельства на разрабатываемое программное обеспечение. Разработанное программное обеспечение включено в Единый реестр российских программ для электронных вычислительных машин и баз данных	– Свидетельство о государственной регистрации программы для ЭВМ в Роспатенте; – Приказ Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации о включении сведений о программном обеспечении в Единый реестр российских программ для электронных	

№	Описание мероприятия	Результат (в т.ч. количественно измеримый)	Подтверждающие документы	Сроки реализации Проекта
			вычислительных машин и баз данных	

5.4 Требования к организациям-соисполнителям работ

Привлечение соисполнителей для целей разработки Решения не предусмотрено.

6 Порядок контроля и приемки результатов работ по разработке Решения

6.1 Виды, состав и методы испытаний разработанного Решения и его составных частей

6.1.1 Требования к видам испытаний Решения

Для проверки выполнения функций программного обеспечения Решения устанавливаются следующие виды испытаний:

- опытная эксплуатация;
- приемочные испытания.

Проверка работоспособности программного обеспечения Решения, включая проведение приемочных испытаний, должна осуществляться на тестовом стенде Исполнителя. Требования к архитектуре, техническому и программному обеспечению тестового стенда представлены в разделе 4.4.3 и 4.4.5.

По итогам проверки Решения оформляется Акт о завершении опытной эксплуатации и готовности Решения к приемочным испытаниям.

Приемочные испытания проводятся с целью установления соответствия представленных Исполнителем выполненных работ на соответствие требованиям Технического задания.

7 Требования к документированию

7.1 Перечень документов, предъявляемых по окончании соответствующих этапов работ

На первом этапе Проекта должен быть подготовлен комплект следующих документов:

- Детализированное описание модели монетизации Решения;
- Алгоритмическая документация, содержащая описание используемых в Решении математических методов;
- Презентация концепции реализации Проекта.

На втором этапе выполнения Проекта должен быть подготовлен комплект следующих документов:

- ведомость технического проекта;
- пояснительная записка к техническому проекту;

На третьем этапе выполнения Проекта должен быть подготовлен комплект следующих документов:

- руководство пользователя Решения;
- руководство администратора Решения;
- руководство для интегратора, содержащее рекомендации по развертыванию и интеграции Решения, организации мониторинга его работоспособности;
- программа и методика приемочных испытаний;
- акт о завершении опытной эксплуатации и готовности Решения к приемочным испытаниям;

– полный актуальный дистрибутив Решения, записанный на внешний носитель;

– протокол приемочных испытаний;

– свидетельство о государственной регистрации программы для ЭВМ в Роспатенте;

– приказ Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации о включении сведений о программном обеспечении в единый реестр российских программ для электронных вычислительных машин и баз данных.

На английский язык должны быть разработаны и переведены следующие технические и эксплуатационные документы:

- архитектура (дизайн) программного обеспечения (software architecture design document);
- документация по API (API documentation);
- руководство конечного пользователя (end-user guide);
- руководство системного администратора (system administration documentation);
- инструкция по установке (installation instruction).

8 Требования к регистрации ПО Решения

По итогам разработки Решения Исполнитель должен обеспечить государственную регистрацию Решения в Роспатенте. Также Исполнитель обязан провести регистрацию Решения в Едином реестре российских программ для электронных вычислительных машин и баз данных.

По итогам регистрации должны быть получены следующие документы:

- Свидетельство государственной регистрации программы для ЭВМ в Роспатенте;
- Приказ Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации о включении сведений о программном обеспечении в единый реестр российских программ для электронных.

9 Источники разработки

Источниками для разработки настоящего детализированного Технического задания являются:

- Описание аналитической информационной системы «Визари».
- Детализированное описание модели монетизации Решения.