

УТВЕРЖДАЮ

Генеральный директор

ООО «НПЦ «БизнесАвтоматика»

П.С. Петраков

«13» 03 2023 г.



АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ПРОЕКТНОГО УПРАВЛЕНИЯ «ВИЗАРИ ПРОЕКТ»

АИС «ВИЗАРИ ПРОЕКТ»

Детализированное техническое задание

На 55 листах

Москва



Содержание

Термины и сокращения.....	4
1 Общие сведения.....	6
1.1 Полное наименование системы и ее условное обозначение	6
1.2 Перечень документов, на основании которых ведется разработка Решения	6
1.3 Сроки выполнения работы.....	6
2 Назначение и цели разработки Решения	7
2.1 Назначение Решения	7
2.2 Цели создания Решения	7
3 Характеристика объекта автоматизации	9
4 Требования к Решению	10
4.1 Требования к структуре Решения в целом	10
4.1.1 Требования к архитектуре.....	10
4.1.2 Перечень подсистем.....	11
4.1.3 Требования к способам и средствам связи между компонентами Решения	14
4.1.4 Требования к режимам функционирования Решения	14
4.1.5 Требования по диагностированию Решения	15
4.1.6 Перспективы развития, модернизации Решения	15
4.2 Требования к функциям (задачам) Решения	16
4.2.1 Функциональные требования к подсистеме управления проектами	16
4.2.2 Функциональные требования к подсистеме нормативно-справочной информации.....	19
4.2.3 Функциональные требования к интеграционной шине	21
4.2.4 Функциональные требования к подсистеме предобработки информации ETL	23
4.2.5 Функциональные требования к подсистеме автоматизированного проектирования предметной области.....	25
4.2.6 Функциональные требования к подсистеме защищенного распределенного хранения данных.....	27
4.2.7 Функциональные требования к подсистеме управления регламентированными процессами	28
4.2.8 Функциональные требования к подсистеме управления задачами.....	31
4.2.9 Функциональные требования к подсистеме уведомлений	35
4.2.10 Функциональные требования к подсистеме поиска	35
4.2.11 Функциональные требования к подсистеме безопасности	37
4.2.12 Функциональные требования к подсистеме администрирования.....	38
4.3 Требования к видам обеспечения Решения.....	38
4.3.1 Требования к математическому обеспечению	38
4.3.2 Требования к информационному обеспечению	39
4.3.3 Требования к лингвистическому обеспечению	39
4.3.4 Требования к программному обеспечению	40
4.3.5 Требования к техническому обеспечению	40
4.4 Общие технические требования.....	41
4.4.1 Требования к численности и квалификации персонала и пользователей Решения	41
4.4.1 Требования к показателям назначения	42
4.4.2 Требования к надежности	43
4.4.3 Требования по безопасности.....	44
4.4.4 Требования к эргономике и технической эстетике.....	45
4.4.5 Требования к защите информации	47
4.4.6 Требования по стандартизации и унификации	48
5 Состав и содержание работ по разработке Решения.....	49

6	Порядок разработки Решения	51
6.1	Порядок организации разработки	51
6.2	Перечень документов и исходных данных для разработки.....	51
6.3	Перечень документов, предъявляемых по окончании соответствующих этапов работ	51
6.4	Требования к гарантийным обязательствам разработчика.....	51
7	Порядок контроля и приемки Решения	52
8	Требования к документированию	53
8.1	Перечень документов, предъявляемых по окончании соответствующих этапов работ	53
9	Требования к регистрации Решения	54
10	Источники разработки	55

Термины и сокращения

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

Таблица 1 – Термины и определения

Термин	Определение
АИС «Визари Project», Решение, Система	Автоматизированная информационная система проектного управления «Визари Project»
Исполнитель	ООО «Научно-производственный центр «БизнесАвтоматика», НПЦ «БизнесАвтоматика»
Проект	Комплекс мероприятий, направленных на разработку (доработку) Решения
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	Расшифровка
АИС	Автоматизированная информационная система
БП	Бизнес-процесс
ЕСИА	Единая система идентификации и аутентификации
ИТ	Информационные технологии
НСИ	Нормативно-справочная информация
ОС	Операционная система
ПО	Программное обеспечение
СМЭВ	Система межведомственного электронного взаимодействия
СОА	Сервисно-ориентированная архитектура
СУБД	Система управления базой данных
ЭВМ	Электронная вычислительная машина
API	(от англ. Application Programming Interface) – набор способов и правил, по которым информационные системы производят обмен данными
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

Сокращение	Расшифровка
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
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
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 Полное наименование системы и ее условное обозначение

Полное наименование системы: Автоматизированная информационная система проектного управления «Визари Project».

Условное обозначение: АИС «Визари Project», далее также – Решение, Система.

1.2 Перечень документов, на основании которых ведется разработка Решения

- Техническое задание на разработку автоматизированной системы проектного управления «Визари Project»;
- Функциональные требования и спецификация программного обеспечения системы проектного управления «Визари Project».

1.3 Сроки выполнения работы

Начало работ: 17.02.2023г.

Окончание работ: 25.03.2024

2 Назначение и цели разработки Решения

АИС «Визари Project» – это кроссплатформенная веб-ориентированная информационная система, которая предоставляет организациям структуру для всех аспектов процесса управления проектами. Система должна автоматизировать процессы управления проектной деятельностью Заказчика, в т. ч. предоставлять единое информационное пространство для участников проектной деятельности, обеспечить удобство коммуникаций, предоставлять доступ к полной информации по реализуемым проектам, обеспечить ускорение сбора информации в формализованном электронном виде по проектной деятельности Заказчика, ее агрегирование и анализ, снижая трудоемкость сбора и анализа, а также обеспечивая дополнительные аналитические возможности.

АИС «Визари Project» в базовой комплектации (комплекс базовых программ) устанавливается в одном, а при необходимости резервирования (повышения отказоустойчивости) и распределения нагрузки в нескольких дата-центрах, в которых осуществляется обучение сотрудников предприятия (организации). АИС «Визари Project» может также включать в себя дополнительные подсистемы, работающие в рамках организационной структуры предприятия (организации) (например, для целей управления).

АИС «Визари Project» обеспечивает для предприятия (организации) единое информационное пространство и гарантирует, что эта информация будет доступна на всех функциональных уровнях иерархии и управления.

2.1 Назначение Решения

Решение предназначено для повышения эффективности деятельности структурных подразделений Заказчика в части реализации ими своих полномочий и функций в сфере управления проектами, программами и портфелями проектов, а также в части управления задачами.

Решение предназначено для:

- Автоматизации деятельности по управлению проектами, портфелями программами.
- Контроля за сроками, бюджетом, целями и показателями проектов, портфелей и программ.
- Ведения проекта, включая управления сроками, ресурсами, бюджетом, целями и показателями.
- Ведения портфелей проектов, в том числе контроля за сроками выполнения проектов, входящих в портфель проектов.
- Формирования аналитических отчетов (дашбордов) и отчетов по ходу реализации проектов.
- Контроля за выполнением результатов.
- Ведение коммуникаций между участниками проектной деятельности.
- Ведение базы знаний по проектам.
- Управление рисками проектов.

2.2 Цели создания Решения

Цель проекта: получить единую информационную систему управления проектами с визуализированным планированием, контролем и отчетностью о ходе выполнения проектов, программ и портфелей проектов.

Ключевыми целями внедрения Решения являются:

- Обеспечение Заказчика инструментом планирования, контроля и отчетности.
- Повышение вовлеченности участников процессов Заказчика в проектную деятельность.
- Облегчение коммуникации участников проекта с помощью создания единой информационной среды для возможности ведения онлайн дискуссий и организации горизонтальных коммуникаций.
- Обеспечение пользователей, которым требуется принятие решений, своевременной и полной информацией.
- Унификацию механизмов управления проектами Заказчика.
- Обеспечение единой точки входа для предоставления отчетных данных по проекту.
- Повышение точности планирования проектов.
- Повышение эффективности управления использованием временных, человеческих и финансовых ресурсов, выделяемых на проекты.
- Повышение удобства работы с большим массивом документов, формируемых в результате проектной деятельности.

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 (Microsoft Project, Oracle Primavera P6, Wrike, Asana, SAP PPM, а также составит конкуренцию наиболее распространенным отечественным системам (Naumen Project Ruler, Directum Projects, ПланФикс, Periscope, ALMI Project Server, 1C:ERP+PM Управление проектной организацией, Реестр проектов (от компании ADVANTA).

3 Характеристика объекта автоматизации

Объектом автоматизации являются процессы управления проектной деятельностью Заказчика, включая информационное пространство участников проектной деятельности, коммуникации участников, сбор информации по проектной деятельности Заказчика, ее агрегирование и анализ, контроль и отчетность о ходе выполнения проектов, программ и портфелей проектов.

4 Требования к Решению

4.1 Требования к структуре Решения в целом

4.1.1 Требования к архитектуре

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

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).

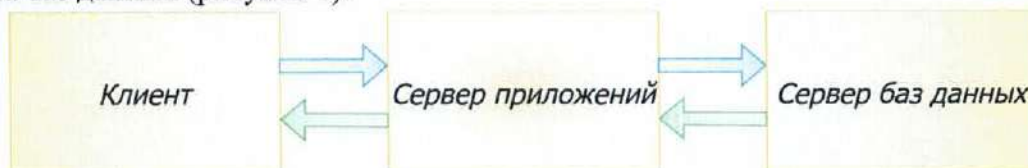


Рисунок 1 – Трехзвенная клиент-серверная архитектура

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.

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

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

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.

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.

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 Решения не должна 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).

4.1.2 Перечень подсистем

Решение должно иметь следующую структуру:

- Подсистема управления проектами;
- Подсистема нормативно-справочной информации;
- Интеграционная шина;
- Подсистема преобразования информации ETL;
- Подсистема автоматизированного проектирования предметной области;

- Подсистема защищенного распределенного хранения данных с встроенными редакторами документов;
- Подсистема управления регламентированными процессами;
- Подсистема управления задачами;
- Подсистема уведомлений;
- Подсистема поиска;
- Подсистема безопасности;
- Подсистема администрирования.

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.

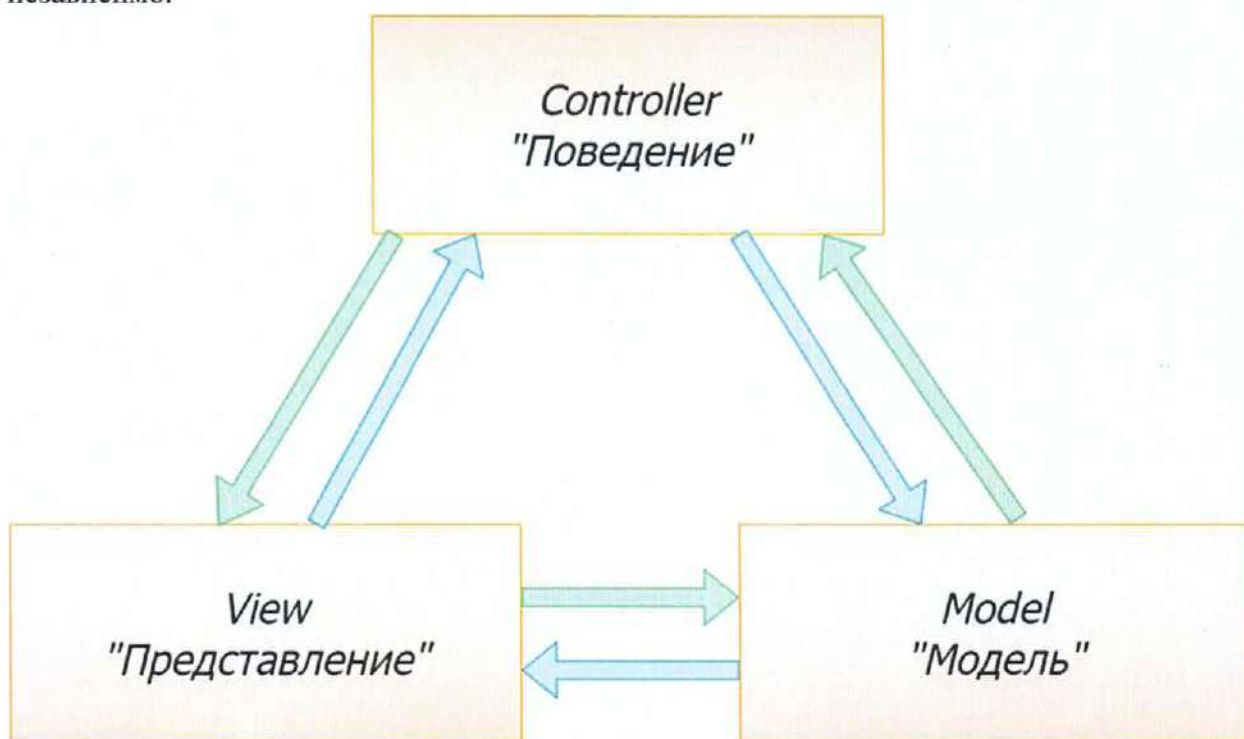


Рисунок 2 – Шаблон проектирования «Модель-Представление-Контроллер»

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).

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

За счёт высокой эффективности масштабирования СОА 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, коммуникационных сервисов и т.д.

СОА 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 компонентов 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.

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);
 - 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/Postgres 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).

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.

4.1.3 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.

4.1.4 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 компонентов;
- 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.

4.1.5 Требования по диагностированию Решения

Решение должно быть снабжено 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 Решении диагностированию подлежат:

- отклонения от заданных параметров быстрого действия 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 подсистем и модулей (компонентов) Решения;
- осуществление резервного копирования и обеспечения безопасности.

4.1.6 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.

4.2 Требования к функциям (задачам) Решения

4.2.1 Функциональные требования к подсистеме управления проектами

В состав подсистемы управления проектами должны быть включены функциональные модули:

- Управление проектами;
- Управление портфелями;
- Управление программами;
- Управление сроками (диаграмма Ганта);
- Управления рисками;
- Управление ресурсами;
- Хранение проектных документов.

4.2.1.1 Управление проектами

Модуль управления проектами должен включать инструменты администрирования проектов.

Модуль должен позволять вести учет проектов, представленных в виде табличного реестра с возможностью перехода в карточку отдельного проекта.

В реестре проектов должно быть предусмотрено наличие панели инструментов, позволяющей проводить следующие операции с проектом (экземпляром объекта):

- Создание нового проекта;
- Редактирование выбранного экземпляра объекта;
- Удаление;
- Перемещение;
- Сортировка;
- Обновление карточки / реестра экземпляров объектов;
- Фильтрация;
- Экспорт в формате MS Excel, PDF;
- Настройка количества отображаемых колонок, содержащих информацию по проекту;
- Просмотр удаленных экземпляров объекта (проектов).

Реестр проектов должен отображать следующую информацию:

- Наименование проекта (ссылка, при нажатии на которую должно открываться окно просмотра основных сведений о проекте);
- Статус проекта;
- Процент выполнения проекта;
- Приоритет;
- Ответственный исполнитель;
- Плановые сроки начала и окончания проекта;
- Наличие разработанного реестра рисков.

Должна быть предусмотрена возможность добавления нового проекта (шаблона объекта) в ручном режиме. При создании проектов на основании предложений в

зависимости от классификации проекта должен использоваться свой шаблон проекта, который определяет правила создания проекта, в том числе перенос из предложения задач.

Должна быть предусмотрена возможность загрузки реестра проектов из формата **xlsx** по установленному шаблону. При этом должна загружаться следующая информация:

- Наименование проекта;
- Описание проекта;
- Запланированная дата начала проекта;
- Запланированная дата окончания проекта.
- Реестр проектов должен иметь средства выгрузки в формат **xlsx**, **pdf**.

Карточка проекта (экземпляра объекта) должна содержать панель инструментов, позволяющую проводить следующие операции с экземпляром объекта (проектом):

- Переход к предыдущему / последующему проекту;
- Копирование ссылки на проект;
- Переход на дашборд (в случае, если информация по проекту подразумевает наличие информации, выводимой на дашборд);
- Печать информации по проекту, при необходимости;
- Просмотр отчетов по проекту (при их наличии).

Карточка проекта должна содержать вкладки, позволяющие переходить в другие модули подсистемы.

4.2.1.2 Управление портфелями

В подсистеме должен быть реализован модуль управления портфелями. Должна быть предусмотрена возможность отображения всех портфелей, проектов и программ в виде списка с возможностью перехода в карточку отдельного проекта или программы.

Модуль управления портфелями должен поддерживать функциональные возможности по балансировке портфеля посредством ранжирования и отбора проектов/программ.

Должна быть реализована возможность отображения следующей основной информации по каждому проекту/программе, входящему (входящей) в портфель, а именно: наименования, приоритета выполнения, руководителя проекта/программы, периода реализации, статуса реализации, статуса отклонения от сроков и иных показателей. Руководителю портфеля должна быть предоставлена возможность внесения изменений в параметры портфеля.

Должна быть реализована возможность просмотра отчетности по портфелю.

Доступ к выбору портфеля и редактированию списка портфелей осуществляется в соответствии с ролевой моделью.

Один проект/программа может быть добавлен только в один портфель.

Отдельные проекты/программы могут внедряться без добавления в портфель.

Портфель проектов должен содержать:

- Иерархический список проектных портфелей/программ/проектов;
- Список проектов, относящихся к определенному портфелю.

4.2.1.3 Управление программами

В подсистеме управления проектами должен быть реализован модуль управления программами проектов. Функциональные возможности модуля должны позволять

создавать программы проектов с перечислением следующей информации по программе проектов:

- Наименование;
- Описание;
- Цели;
- Этап программы;
- Руководитель программы;
- Подразделение;
- Период реализации программы;
- Файлы.

Должна быть реализована возможность ведения истории изменений различных параметров реализации программы.

Должна быть реализована возможность перехода из табличной формы проекта, в отдельную карточку параметра программы.

4.2.1.4 Управление сроками (диаграмма Ганта)

Для визуализации сроков проекта должна быть реализована диаграмма Ганта.

В левом блоке должна отображаться информация по проекту в текстовом виде: наименование проекта, дата начала, дата окончания, статус, исполнитель, процент выполнения.

Модуль должен интегрироваться с подсистемой управления задачами.

При выборе проекта должна быть возможность раскрытия ветки проекта и вывода перечня задач по проекту.

Пользователь должен иметь возможность устанавливать связи между задачами непосредственно на диаграмме Ганта.

4.2.1.5 Управления рисками

Модуль управления рисками должен обеспечивать возможность работы с реестром рисков проектов. Реестр рисков в табличной форме должен содержать следующие параметры:

- Наименование проекта (программы, портфеля);
- Наименование риска;
- Вид риска;
- Факторы риска;
- Последствия риска;
- Область риска;
- Решение о реагировании на риск;
- Дата принятия решения о реагировании на риск.

При работе с реестром рисков должна быть реализована возможность настройки последовательности отображения полей табличного реестра, добавления новых полей (из числа параметров учета карточки риска), скрытия видимости полей.

При работе с реестром рисков пользователю, обладающему соответствующими правами, должна быть предоставлена возможность:

- Создания нового риска;
- Копирования риска;
- Удаления риска из реестра;

- Настройки сортировки рисков;
- Настройки фильтрации рисков;
- Обновления реестра рисков;
- Экспорта реестра рисков в excel, pdf;
- Переключение режима на просмотр записей, удаленных из реестра рисков.

Модуль управления рисками должен обеспечивать возможность формирования карточки нового риска.

В карточке риска должна быть обеспечена возможность ведения реестра изменений различных параметров риска.

Пользователь должен иметь возможность мониторинга реализации риска путем указания статуса риска.

Должна быть возможность ведения справочника типовых рисков проектов, возможность формулировки риска проекта на основе справочника.

4.2.1.6 Управление ресурсами

Модуль должен предоставлять возможность управления человеческими ресурсами, задействованными в проекте.

По каждому проекту должна формироваться команда (реестр) из пользователей Решения.

Карточка ресурса должна содержать поля:

- Пользователь (ФИО);
- Роль на проекте:
 - Владелец;
 - Руководитель проекта;
 - Администратор проекта;
 - Исполнитель;
 - Наблюдатель;
- Активный (флажок);
- Ответственность (флажок).

Пользователь должен иметь возможность связывать участников проекта с существующими пользователями, а также вводить новых участников.

4.2.1.7 Хранение проектных документов.

Модуль хранения проектных документов должен обеспечивать хранение документации по проекту, а также файлов любого формата, относящихся к проекту.

Модуль должен быть интегрирован с подсистемой защищенного распределенного хранения данных и должен позволять загружать документы как их хранилища, так и с персонального компьютера.

Модуль должен также позволять вводить документы вручную т. е. должен предоставлять встроенный редактор документов.

Должна быть предоставлена возможность интеграции модуля с подсистемой управления регламентированными процессами при необходимости согласования проектных документов.

4.2.2 Функциональные требования к подсистеме нормативно-справочной информации

Подсистема должна 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.

Подсистема справочников и классификаторов должна позволять добавлять новые записи (типы и виды объектов), создавать многоуровневые иерархические структуры реестров, справочников и классификаторов.

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

При просмотре удаленных записей должна быть возможность восстановить конкретную запись.

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).

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.

Требуется провести интеграцию подсистемы в архитектуру Решения.

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

4.2.3 Функциональные требования к интеграционной шине

Решение должно включать предустановленные интеграции.

Интеграционная шина должна поддерживать три основных механизма интеграции:

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

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

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

Должны быть 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.

4.2.3.1 Требования к функциям модуля 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 предобработки 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.

4.2.4 Функциональные требования к подсистеме предобработки информации ETL

Наличие подсистемы предобработки 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.

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

С помощью ETL-сервиса 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.

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 должен предоставлять набор готовых сценариев и мастеров и состоять из следующих пунктов меню:

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

Должен быть разработан интерфейс XSLT-схем. В верхней части карточки XSLT-схемы (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 карточки должны располагаться кнопки команд «Сохранить» и «Сохранить и закрыть».

Должны быть разработаны обучающие материалы, обеспечивающие методологическую поддержку пользователей.

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

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 подсистему 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.

4.2.5 Функциональные требования к подсистеме автоматизированного проектирования предметной области

Подсистема автоматизированного проектирования предметной области должна быть построена на базе конструктора объектного проектирования (low-code) с возможностью кодогенерации классов.

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

Подсистема автоматизированного проектирования предметной области должна 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:

- а) ввод параметров нового класса (представления модели данных):
 - основных данных:
 - имени;
 - наименования;
 - параметров сборки;
 - параметров использования в регламентированных процессах;
 - 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;
- б) трансляция параметров модели в машинный формат;
- в) формирование табличного представления объекта в базе данных;
- г) сборка связей, внешних и внутренних ссылок в базе данных;
- д) настройка прав доступа и схем движения объектов класса.

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

Конструктор объектного проектирования 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.

Должна быть проведена интеграция подсистемы автоматизированного проектирования предметной области в общую архитектуру Решения.

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

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

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

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.

4.2.6 Функциональные требования к подсистеме защищенного распределенного хранения данных

Подсистема защищенного распределенного хранения данных должна 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.

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

Подсистема должна обеспечивать резервное копирование данных и сохранение их копий по физически распределенным кластерным узлам.

Подсистема защищенного распределенного хранения данных с встроенными редакторами документов в Решения должна быть интегрирована в единую архитектуру Решения.

Подсистема должна 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.

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.

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;
- настройку отображения нескольких папок на диске или другой папке в виде галереи отдельных иконок.

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».

Подсистема защищенного распределенного хранения данных должна 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.

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

4.2.7 Функциональные требования к подсистеме управления регламентированными процессами

Подсистема управления регламентированными процессами (BPM), в основе которого лежит Workflow-конструктор, должна обеспечивать гибкие возможности по оперативной

настройке регламентированных процессов, в том числе процессов формирования, согласования и утверждения отчетности и иных сопутствующих материалов.

Конструктор автоматизации регламентированных процессов (Workflow) должен реализовать 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 подсистемы управления регламентированными процессами, определяющей базовый набор сущностей и атрибутов для описания процессов, является 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.

Прототип интерфейса конструктора Workflow представлен на рисунке 3.

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

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

Панель инструментов должна содержать следующие элементы:

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

— Увеличить масштаб.

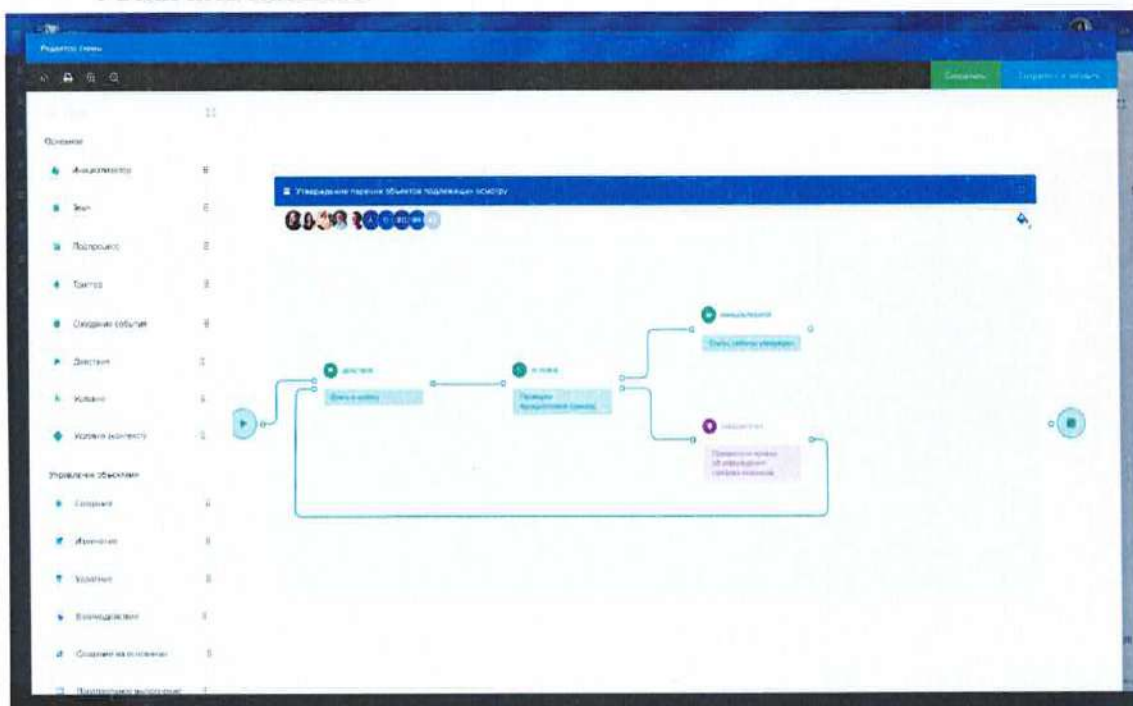


Рисунок 3 – Прототип интерфейса конструктора Workflow

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

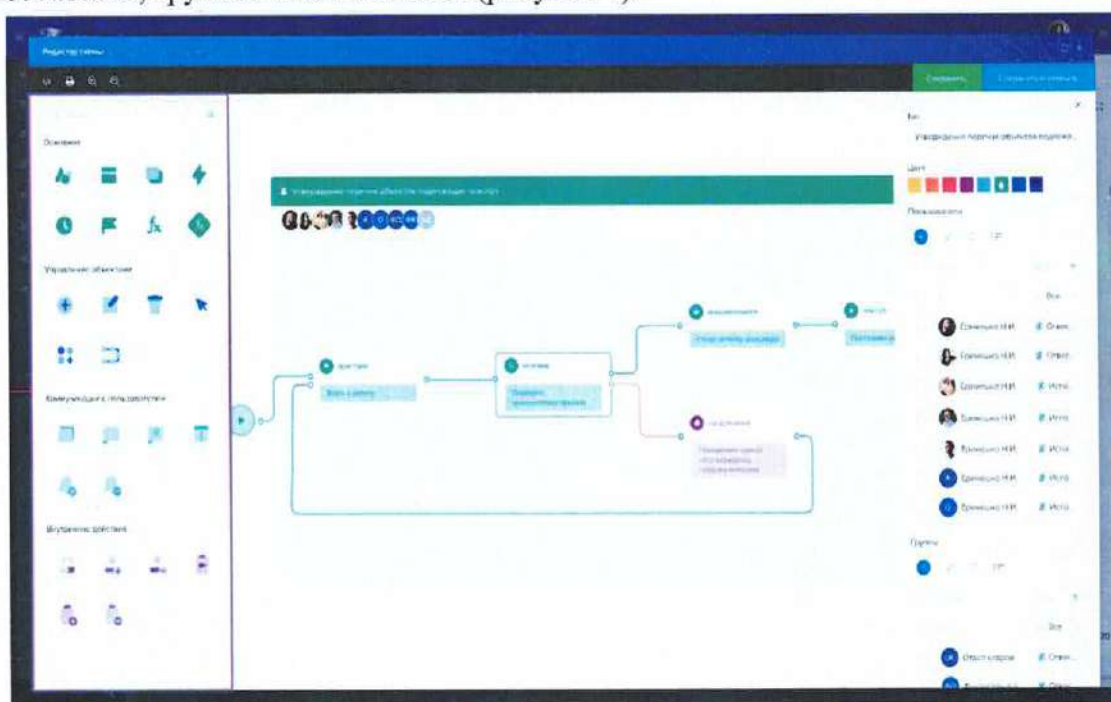


Рисунок 4 – Прототип интерфейса конструктора Workflow с панелью настроек, содержащую информацию о пользователях

Должна быть проведена интеграция подсистемы с остальными структурными элементами Решения для формирования универсального решения многомерного анализа и визуализации больших данных.

4.2.8 Функциональные требования к подсистеме управления задачами

Подсистема управления задачами предназначена для автоматизации процессов постановки задач и контроля их исполнения.

Должен быть сформирован банк карточек задач в зависимости от категории задачи. Задачи должны быть подразделены на следующие категории:

- а) По статусу задачи:
 - новая,
 - назначена,
 - в работе,
 - на тестировании,
 - на согласовании,
 - согласована,
 - отклонена,
 - решена;
- б) По исполнителю:
 - я исполнитель,
 - мои задачи на исполнении;
- в) По автору;
- г) В зависимости от роли в команде:
 - куратор,
 - наблюдатель,
 - соисполнитель,
 - соавтор;
- д) По приоритету:
 - высокий,
 - средний,
 - низкий;
- е) По типу задачи:
 - аналитика,
 - дизайн,
 - разработка/

Подсистема должна быть 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.

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

Подсистема управления задачами должна 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).

Подсистема управления задачами должна 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.

Подсистема управления задачами должна обеспечивать следующие возможности:

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

- выбора ответственного исполнителя среди пользователей системы. Должна быть 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.

- возможность цветовой дифференциации любым цветом из матрицы RGBA;

- возможность ведения статусов задач.

- возможность ведения истории задачи с указанием исполнителей, дат совершения действий над задачей, а также комментария пользователя;

- возможность прикрепления файлов любых форматов к задаче.

Подсистема управления задачами должна интегрироваться с подсистемой управления регламентированными процессами и подсистемой 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.

Должна быть проведена интеграция подсистемы в архитектуру Решения.

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

4.2.8.1 Алгоритм движения задачи

Подсистема управления задачами должна обеспечивать возможность четкого отображения процесса перемещения задачи между этапами и механизмы, влияющие на выбор траектории перемещения под воздействием внешних и внутренних факторов. Подсистема должна обеспечивать визуализацию зависимостей между задачами для чего необходимо обеспечить соответствие схемы следующей математической модели.

Схема движения S задачи представляет собой тройку $S =$, 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 $C_i =$, 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 5.

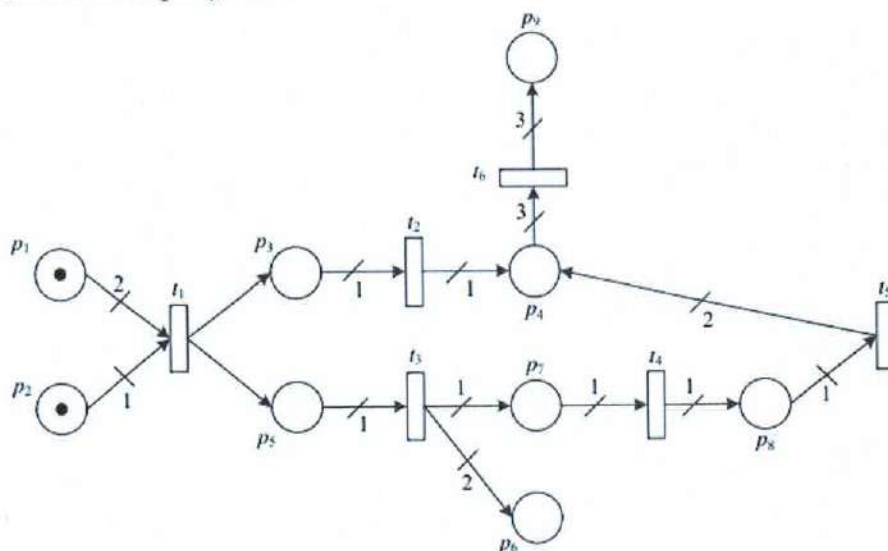


Рисунок 5 – Пример используемой модификации иерархической сети

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:

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

4.2.9 Функциональные требования к подсистеме уведомлений

Подсистема уведомлений должна быть интегрирована с подсистемой управления регламентированными процессами.

Принцип работы подсистемы уведомлений должен быть 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.

Подсистема уведомлений должна обеспечивать 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 должно автоматически переноситься в архив.

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

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 бизнес-процессов или о других различных событиях используются настройки почтового сервера.

4.2.10 Функциональные требования к подсистеме поиска

Подсистема 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

осуществляется по всем материалам, открытым для поиска пользователю с учетом связанности информации.

В режиме расширенного поиска подсистема должна 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.

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

Подсистема поиска должна позволять осуществлять каскадный поиск (поиск в результатах поиска).

При отображении списков данных должны быть 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 подсистемы поиска в форматах xls, xlsx;
- индивидуальная настройка параметров отображения для каждого пользователя.

Подсистема поиска должна 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 подсистеме поиска должна быть предусмотрена настройка перечня выводимых на экран полей таблицы для каждого пользователя (индивидуальный поиск).

Подсистема поиска должна позволять настраивать фильтры не только для отображаемых полей, но и для скрытых полей мнемоник, а также связанных объектов.

Необходимо 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, должно быть представлено поле выбора фильтра.

При создании фильтра пользователем должны задаваться следующие параметры:

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

Должны быть реализованы следующие функции:

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

Должна быть проведена интеграция подсистемы поиска в общую архитектуру Решения.

4.2.11 Функциональные требования к подсистеме безопасности

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

Подсистема безопасности должна 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.

Подсистема безопасности должна 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.

Подсистема безопасности должна представлять механизмы:

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

Подсистема безопасности должна обеспечивать:

- возможность создания групп доступа и групп пользователей с заданными правами доступа, к которым относятся:
 - полный доступ к внутренним и внешним разделам 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 прототипа Решения (обеспечение доступа только зарегистрированных пользователей, прошедших процедуру идентификации и аутентификации);
- регистрацию (журналирование) входа субъектов доступа в Систему на базе 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 подсистемы 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.

4.2.12 Функциональные требования к подсистеме администрирования

Подсистема администрирования должна позволять администратору настраивать параметры функционирования ПО в целом или отдельных компонентов:

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

Подсистема администрирования должна обеспечивать удаленное журналирование операций, в том числе действий пользователей.

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

Подсистема администрирования должна 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 Требования к видам обеспечения Решения

4.3.1 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.

4.3.2 Требования к информационному обеспечению

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.

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.

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 компонентами Решения.

С точки зрения доступа 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.

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

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

Для обмена данными с внешними должны быть использованы следующие адаптеры:

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

4.3.3 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.

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

4.3.4 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.

4.3.5 Требования к техническому обеспечению

Решение может быть установлено и запущено 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.

4.4 Общие технические требования

4.4.1 Требования к численности и квалификации персонала и пользователей Решения

Численность персонала определяется количеством пользователей Решения.

Пользователи должны обладать навыками работы на компьютере.

Администраторы Решения должны обладать навыками администрирования информационных систем.

Администраторы Решения должны обеспечивать функционирование подсистем и 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 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.

4.4.1 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.

4.4.2 Требования к надежности

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.

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.

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

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

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.

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.

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.

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.

4.4.3 Требования по безопасности

4.4.3.1 Требования к условиям окружающей среды

Требования к условиям окружающей среды не предъявляются.

4.4.3.2 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.

4.4.3.3 Требования по электропитанию и заземлению

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

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.

4.4.3.4 Требования к противопожарной безопасности

Требования к противопожарной безопасности не предъявляются.

4.4.4 Требования к эргономике и технической эстетике

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

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

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.

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.

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.

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.

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.

Должна быть обеспечена возможность масштабирования экранных форм.

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.

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

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.

4.4.5 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.

4.4.6 Требования по стандартизации и унификации

Должно быть обеспечено соответствие Решения требованиям следующих стандартов:

- HTTP 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;
- подсистемы защищенного хранилища;
- подсистемы интеграционной шины.

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.

5 Состав и содержание работ по разработке Решения

Этапы и сроки работ по разработке Решения представлены в таблице 4.

Таблица 4 – Этапы и сроки работ по разработке Решения

№	Описание мероприятия	Результат (в т.ч. количественно измеримый)	Подтверждающие документы	Сроки
Этап 1				
1.1	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	С 17.02.2023 по 13.03.2023
Этап 2				
2.1	Разработка подсистем Решения	Проведена разработка Решения согласно требованиям Детализированного технического задания, а именно разработка подсистем: – управления проектами; – нормативно-справочной информации; – интеграционной шины; – предобработки информации ETL; – защищенного распределенного хранения данных; – управления регламентированными процессами; – поиска; – безопасности; – администрирования.	– Ведомость технического проекта; – Пояснительная записка к техническому проекту; – Программа и методика предварительных испытаний.	С 14.03.2023 по 04.03.2024
Этап 3				
3.1	Тестирование Решения	Проведено функциональное и нагрузочное тестирование Решения	– Руководство пользователя Решения;	С 05.03.2024 по 25.03.2024

№	Описание мероприятия	Результат (в т.ч. количественно измеримый)	Подтверждающие документы	Сроки
		Разработана пользовательская эксплуатационная документация.	– 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.	
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 вычислительных машин и баз данных	

6 Порядок разработки Решения

6.1 Порядок организации разработки

При выполнении работ Исполнитель должен руководствоваться требованиями стандарта качества ГОСТ Р ИСО 9001-2015, а также учитывать требования следующих стандартов:

- РФ ГОСТ Р ИСО 21500–2014 «Руководство по проектному менеджменту»;
- ГОСТ Р 54869–2011 «Проектный менеджмент. Требования к управлению проектом»;
- ГОСТ Р ИСО/МЭК 12207–2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств».

6.2 Перечень документов и исходных данных для разработки

Перечень документов и исходных данных для разработки представлен в п. 1.2 и разделе 10.

6.3 Перечень документов, предъявляемых по окончании соответствующих этапов работ

Перечень документов, предъявляемых по окончании соответствующих этапов работ представлен в разделе 5.

6.4 Требования к гарантийным обязательствам разработчика

Исполнитель обеспечивает гарантийное обслуживание Системы в течение 12 (двенадцати) календарных месяцев. Началом гарантийного срока обслуживания является дата подписания акта приемки в промышленную эксплуатацию.

Исполнитель должен гарантировать, что разработанное программное обеспечение Системы будет функционировать в соответствии со своим назначением в течение всего гарантийного срока.

В гарантийное сопровождение не входят услуги, связанные:

- с администрированием локальных и глобальных сетей;
- с установкой и администрированием операционных систем и другого общесистемного ПО;
- с установкой и администрированием СУБД;
- с установкой и настройкой новых версий прикладного ПО на оборудование Заказчика;
- с администрированием прикладного ПО на объектах Заказчика;
- с обучением новых сотрудников работе с функционалом прикладного ПО и обучением пользователей функционалу новых версий прикладного ПО.

7 Порядок контроля и приемки Решения

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

- предварительные испытания;
- опытная эксплуатация;
- приемочные испытания.

Проверка работоспособности программного обеспечения Решения, включая проведение предварительных и приемочных испытаний, должна осуществляться на тестовом стенде Исполнителя.

Требования к архитектуре Решения представлено в п. 4.1.1.

Требования к программному обеспечению тестового стенда представлены в п. 4.3.4.

Требования к техническому обеспечению тестового стенда представлены в п. 4.3.5.

Предварительные испытания проводят для определения работоспособности программного обеспечения Решения и определения возможности ввода Решения в опытную эксплуатацию. Испытания должны проводиться согласно документу «Программа и методика предварительных испытаний», разработанному Исполнителем.

По завершении опытной эксплуатации 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.

8 Требования к документированию

8.1 Перечень документов, предъявляемых по окончании соответствующих этапов работ

На первом этапе Проекта должен быть подготовлен комплект следующих документов:

- Алгоритмическая документация, содержащая описание используемых в Решения 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;
- программа и методика предварительных испытаний.

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

- протокол предварительных испытаний;
- 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.

9 Требования к регистрации 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.

10 Источники разработки

Источниками для разработки настоящего детализированного Технического задания являются:

- Описание аналитической информационной системы «Визари».