## Общество с ограниченной ответственностью «Тетракуб» ОГРН 1107847219480 ИНН 7802719547

**УТВЕРЖДАЮ** 

Генеральный директор ООО «Тетракуб» Лексашов Александр Викторович

## Описание функциональных характеристик программного обеспечения «Индустриальная аналитическая информационная система «Indanis»»

2025

# СОДЕРЖАНИЕ

1. C	Общее положение
2. C	Общее описание Программы5
2.1.	Назначение Программы5
2.2.	Состав программы5
3. C	Описание функций программы8
3.1.	Модуль управления пользователями8
3.2.	Модуль управления учетными записями пользователей 8
3.3.	Библиотека для генерации токенов10
3.4.	Модуль управления ролями и правами 11
3.5.	Модуль интеграции с LDAP11
3.6.	Модуль нотификации 12
3.7.	Модуль информации о событиях13
3.8.	Менеджер лицензий13
3.9.	Файловый менеджер14
3.10	Э. Модуль управления группами и проектами
3.1	I. Модуль управления директориями и таблицами16
3.12	2. Модуль управления вычислителями 18
3.13	3. Модуль управления формами параметров 18
3.14	4. Модуль управления расчетами
3.1:	5. Модуль управления дашбордами
3.10	6. Модуль управления приложением
3.1′	7. Вспомогательные модули
3.18	8. Модуль пользовательского интерфейса

3	8.19.	Сервис обработки задач	
4.	Поль	зовательские сценарии	

### 1. Общее положение

Основной целью программного продукта «Индустриальная аналитическая информационная система «Indanis»» является обеспечение возможности управления данными и расчетами компьютерного моделирования производственных и технологических процессов.

Настоящий документ содержит описание функциональных характеристик программного обеспечения «Индустриальная аналитическая информационная система «Indanis»» (далее – Программа).

## 2. Общее описание Программы

## 2.1. Полное наименование Программы

Индустриальная аналитическая информационная система «Indanis»

## 2.2. Назначение Программы

Индустриальная аналитическая информационная система «Indanis» предназначена для управления данными и расчетами компьютерного моделирования производственных и технологических процессов.

## 2.3. Состав программы

Индустриальная аналитическая информационная система «Indanis» состоит из двух основных блоков – веб-приложения и сервиса обработки задач, которые в свою очередь состоят из клиентов и серверов, а также службы каталогов, которая соединена с сервером веб-приложения через безопасный протокол доступа к каталогам LDAPS.

Клиент веб-приложения является модулем пользовательского интерфейса и соединен с сервером веб-приложения посредством REST API и WebSocket API. Сервер веб-приложения состоит ядра, базы данных, локального файлового хранилища, файлового хранилища Minio и SMTPсервера. В состав ядра входит:

- Модуль управления пользователями;
- Модуль управления учетными записями;
- Библиотека для генерации токенов;
- Модуль управления ролями и правами;
- Модуль интеграции с LDAP;
- Модуль нотификаций;
- Модуль информации о событиях;
- Менеджер лицензий;

- Файловый менеджер;
- Модуль управления группами и проектами;
- Модуль управления директориями и таблицами;
- Модуль управления вычислителями;
- Модуль управления формами параметров;
- Модуль управления расчетами;
- Модуль управления дашбордами;
- Модуль управления приложением;
- Вспомогательные модули.

Клиент сервиса обработки задач взаимодействует с серверами сервиса обработки задач через JSON-RPC, при этом серверов в рамках сервиса обработки задач может быть и больше 3 штук. Каждый сервер сервиса обработки задач взаимодействует с сервером веб-приложения через REST API. Архитектура Программы представлена на рисунке 1.



Рисунок 1 – Архитектура Индустриальной аналитической информационной

система «Indanis»

#### 3. Описание функций программы

#### 3.1. Модуль управления пользователями

Модуль управления пользователями отвечает за взаимодействие с пользователями системы и обеспечивает выполнение ключевых операций, таких как создание и удаление пользователей, а также предоставляет инструменты для хранения и управления персональными данными.

Модуль также позволяет управлять профилем пользователя. Пользователи могут редактировать свои персональные данные, загружать или удалять аватары. В частности, доступно обновление таких полей, как ФИО, название предприятия, подразделение, должность, адрес электронной почты, контактный номер телефона, и т.д. Это помогает поддерживать актуальность информации в системе.

Для администраторов предусмотрены расширенные инструменты управления пользователями. Им доступна задание формата адреса электронной почты, используемых при регистрации. Это необходимо для обеспечения соответствия определенным требованиям, например, чтобы в системе могли зарегистрироваться только пользователи, чьи адреса соответствуют шаблону корпоративной почты. В случаях, когда e-mail не соответствуют требованиям, администратор может подтверждать или отклонять дальнейший процесс регистрации (в рамках Модуля управления учетными записями пользователей). Также администраторам доступен просмотр детальной информации обо всех пользователях, что обеспечивает контроль и прозрачность процессов управления.

#### 3.2. Модуль управления учетными записями пользователей

Модуль управления учетными записями пользователей предназначен для обеспечения эффективного и безопасного управления процессами

аутентификации и защиты учетных записей пользователей. Он реализует набор функций, направленных на создание, модификацию и контроль учетных записей, а также внедряет меры для обеспечения безопасности данных.

Модуль предоставляет возможность регистрации учетных записей пользователей. После перехода по ссылке активации и установки пароля введенные данные шифруются с использованием современных алгоритмов хэширования, обеспечивая безопасное хранение.

Одной из ключевых функций является обеспечение безопасной аутентификации пользователей, включая проверку логинов и паролей, а также поддержку различных методов верификации. Пользователям доступны инструменты для сброса, изменения и восстановления паролей, что упрощает процесс управления доступом к системе.

Безопасность учетных записей обеспечивается за счет внедрения строгих требований к паролям, таких как минимальная и максимальная длина, обязательное использование символов, заглавных букв и цифр, а также регулярной смены паролей. Кроме того, система поддерживает политики безопасности, включающие ограничение количества неудачных попыток входа и временную блокировку учетной записи при превышении установленного лимита попыток.

Администраторам предоставляется возможность редактировать настройки безопасности учетных записей пользователей, адаптируя их под изменяющиеся требования безопасности Модуль системы. также поддерживает мониторинг активности пользователей, включая отслеживание последних входов в систему, историю изменений паролей и выявление подозрительных попыток входа. Для повышения безопасности реализовано автоматическое завершение сессий после определенного времени бездействия.

Таким образом, модуль управления учетными записями пользователей играет важную роль в обеспечении безопасного доступа пользователей к системе, защите данных и предотвращении несанкционированного использования учетных записей.

#### 3.3. Библиотека для генерации токенов

Библиотека генерирует Access-токены, которые используются для аутентификации пользователя и предоставления ему доступа к системе. Эти токены являются ключевым элементом безопасности, позволяя пользователю взаимодействовать с системой на основе предоставленных прав.

Также библиотека генерирует Refresh-токены, которые продлевают срок действия Access-токенов без необходимости повторной аутентификации. Это позволяет пользователю оставаться в системе длительное время, обеспечивая удобство использования при сохранении высокого уровня безопасности.

Кроме того, библиотека предоставляет возможность генерации сервисных токенов, которые используются для интеграции внешних сервисов с системой. При создании таких токенов указывается определенный набор прав, определяющий уровень доступа сервиса к функционалу системы.

Все токены имеют ограниченный срок действия, что значительно повышает уровень безопасности системы. Ограничение времени жизни токенов минимизирует риски кражи или компрометации данных, так как даже если токен будет перехвачен, он станет недействительным через определенный промежуток времени.

Генерация токенов происходит с использованием алгоритма шифрования HS256, который обеспечивает надежную защиту передаваемых

данных. Этот алгоритм гарантирует, что токены невозможно подделать или изменить без соответствующего секретного ключа.

Кроме того, библиотека предоставляет информацию о текущем аутентифицированном пользователе. Это позволяет системе корректно управлять правами доступа и персонализировать взаимодействие с пользователем на основе его роли и параметров учетной записи.

#### 3.4. Модуль управления ролями и правами

Модуль обеспечивает управление правами доступа в системе на основе ролевой модели RBAC (Role-Based Access Control). Он предоставляет возможность гибко настраивать права, привязывать их к ролям и управлять этими ролями для пользователей системы.

Модуль позволяет задавать различные типы прав, такие как доступ на чтение, запись или исполнение. Администратору предоставляется возможность создавать, редактировать и удалять роли.

Модуль предоставляет предопределенные роли, которые имеют фиксированные наборы прав доступа. Они предназначены для упрощения управления правами пользователей, поскольку уже включают необходимые разрешения для типичных сценариев использования.

Таким образом, модуль обеспечивает полный цикл управления ролями и правами доступа, позволяя эффективно адаптировать систему под изменяющиеся требования.

#### 3.5. Модуль интеграции с LDAP

Модуль интеграции с LDAP отвечает за аутентификацию и авторизацию пользователей через внешнюю службу каталогов, такую как OpenLDAP или Active Directory. При входе в систему учетные данные пользователя проверяются через LDAP, после чего система определяет его уровень доступа в соответствии с заданными настройками безопасности.

Если пользователь ранее не входил в систему через LDAP, его учетные данные, полученные из службы каталогов, используются для автоматического создания пользователя в системе. Это позволяет избежать ручной регистрации и упрощает процесс управления пользователями.

При каждом входе в систему через LDAP данные учетной записи пользователя синхронизируются с актуальными данными из службы каталогов. Это гарантирует, что информация о пользователе в системе всегда соответствует данным в OpenLDAP или Active Directory.

Если учетная запись пользователя в службе каталогов заблокирована или срок ее действия истек, система не позволит пользователю войти. Это обеспечивает дополнительный уровень безопасности и предотвращает доступ неактивных или заблокированных пользователей.

Подключение к OpenLDAP или Active Directory осуществляется с использованием защищенного протокола LDAPS, обеспечивающего безопасный обмен данными между системой и службой каталогов.

Переключение на аутентификацию через LDAP, а также выбор используемой службы каталогов (OpenLDAP или Active Directory), настраивается в модуле управления приложением. Администратор может задать параметры подключения, определить политику аутентификации и настроить правила синхронизации учетных записей в соответствии с требованиями организации.

#### 3.6. Модуль нотификации

Модуль нотификации обеспечивает отправку уведомлений как внутри системы, так и на электронную почту. Пользователи могут самостоятельно настраивать параметры e-mail-рассылки через свой профиль.

Администраторы имеют возможность управлять правилами отправки уведомлений, включая создание и настройку шаблонов сообщений для различных типов событий.

#### 3.7. Модуль информации о событиях

Модуль автоматически отслеживает и регистрирует все значимые события в системе, включая события, связанные с информационной безопасностью. Каждое событие фиксируется в Журнале с указанием детальной информации: временной метки, описания действия (например, «пользователь зашел под аккаунтом <...>»), данных о пользователе, инициировавшем это действие, критичности события, и т.д.

#### 3.8. Менеджер лицензий

Менеджер лицензий предназначен для управления лицензионными ключами и обеспечения их привязки к конкретному серверу. При установке системы на сервер генерируется уникальный идентификатор, который создается на основе характеристик аппаратного обеспечения (например, серийного номера материнской платы). Этот идентификатор используется для привязки лицензионного ключа к конкретному серверу, что исключает возможность использования одного ключа на нескольких системах.

Лицензионный ключ содержит важную информацию о лицензии, включая название организации, тип лицензии, срок действия, а также ограничения на количество пользователей и проектов. Администраторы могут просматривать данную информацию по каждой лицензии.

Администраторы имеют возможность добавлять новые лицензионные сертификаты. Перед сохранением нового сертификата или просмотром информации об активной лицензии выполняется верификация сертификата. Верифицированный сертификат должен соответствовать определенному формату.

В процессе верификации сертификата выполняются следующие шаги:

- 1. Проверяется подлинность подписи зашифрованных данных в соответствии с указанным алгоритмом подписи.
- 2. Осуществляется расшифровка данных с использованием указанного алгоритма шифрования.
- 3. Проводятся проверки совместимости лицензии.

Кроме того, система автоматически проверяет ограничения активной лицензии перед выполнением определенных операций:

- Перед принятием кода активации или реактивации аккаунта проверяется доступное количество аккаунтов.
- Перед авторизацией пользователя проверяется, не превышено ли ограничение по количеству пользователей.
- Перед созданием нового проекта проверяется лимит доступных проектов.
- Также осуществляется проверка срока действия лицензии, чтобы гарантировать её актуальность.

Таким образом, менеджер лицензий обеспечивает надежное управление лицензионными ключами, защиту от несанкционированного использования и контроль соблюдения ограничений, установленных для каждой лицензии.

#### 3.9. Файловый менеджер

Модуль предоставляет функционал для загрузки и хранения файлов, поддерживая как локальные, так и сторонние файловые хранилища. Это

позволяет гибко организовывать работу с данными в зависимости от потребностей системы.

Модуль предоставляет возможность скачивать файлы из хранилища, что обеспечивает удобный доступ к сохраненным данным. Подобный функционал особенно полезен для обмена информацией между модулями системы.

Модуль поддерживает различные форматы файлов (CSV, XLSX, PNG, JPEG, TIFF, WEBP, ZIP, т.д.). Перед загрузкой файлы проверяются на соответствие расширениям, что помогает избежать ошибок и повышает безопасность системы. Такая проверка гарантирует, что загружаемые данные соответствуют установленным требованиям.

### 3.10. Модуль управления группами и проектами

Модуль управляет группами и проектами, обеспечивая удобную рабочих организацию администрирование пространств. Группы И объединяют связанные проекты виде дерева, позволяет В ЧТО централизованно управлять доступом участников и генерировать сервисные токены. Проекты служат для создания полноценной среды моделирования бизнес-процессов. В рамках проекта можно работать с таблицами, вычислителями, формами параметров, расчетами и отчетами. Это делает модуль универсальным инструментом для анализа данных, планирования и принятия решений.

Модуль предоставляет широкий набор функций. Пользователи могут создавать новые группы и проекты, редактировать их метаданные, такие как название и описание, а также удалять ненужные элементы. Управление участниками включает возможность добавлять и удалять пользователей, назначая им роли с разными уровнями доступа. Кроме того, модуль координирует создание и удаление различных сущностей, таких как папки, таблицы, вычислители, формы параметров, расчеты, шаблоны дашбордов и сервисные токены. Он перенаправляет соответствующие запросы в модули, ответственные за работу с конкретными объектами. После успешного выполнения операции модуль связывает новую сущность с проектом или отвязывает её при удалении, поддерживая целостность структуры проекта.

Дополнительно модуль поддерживает экспорт и импорт проектов, что упрощает перенос данных между системами и создание резервных копий.

#### 3.11. Модуль управления директориями и таблицами

Модуль предназначен для управления директориями и таблицами, предоставляя удобный интерфейс для работы с данными. Директории используются для группировки таблиц в виде дерева, что позволяет организовывать данные логически и создавать иерархическую структуру. Таблицы, в свою очередь, предназначены для хранения и предоставления информации, что делает работу с данными более структурированной и упорядоченной.

Основная цель модуля – предоставить возможность создания и управления таблицами пользователям, не имеющим опыта работы с базами данных. Благодаря удобному интерфейсу, любой человек сможет легко организовывать и хранить свои данные без необходимости изучения сложных технических аспектов.

Модуль позволяет пользователям создавать новые таблицы, а также изменять их структуру в зависимости от потребностей. В процессе редактирования можно:

• Добавлять и удалять столбцы, указывая их ключевые характеристики:

- Название наименование столбца для удобного понимания его назначения.
- Тип данных определение формата хранимой информации (например, текст, число, дата).
- Уникальность возможность запретить дублирование значений в столбце.
- Обязательность требование к заполнению данного столбца (не допускаются пустые значения).
- Порядок в таблице расположение столбца относительно других.
- Добавлять и удалять ограничения на столбцы, указывая:
  - Левую часть условия столбец, к которому применяется ограничение.
  - Оператор сравнения выражение для проверки значений (например, больше или равно, меньше или равно).
  - Правую часть условия конкретное значение или другое поле, с которым идет сравнение.
- Устанавливать связи между таблицами, создавая аналог foreign key.
   Это позволяет формировать логические связи между данными, задавая:
  - Родительскую таблицу и поле-источник указываются данные, на которые будет ссылаться связь.
  - Дочернюю таблицу и связанное поле определяется, какое поле
     в зависимой таблице будет хранить связь с родительской.
  - Тип связи устанавливается зависимость между данными (один к одному, один ко многим).

Для поддержания актуальности данных модуль предоставляет возможность обновления записей в таблицах. Пользователь может редактировать существующие записи, добавлять новые данные или удалять устаревшие строки. Все изменения проверяются на соответствие ограничениям и связям, чтобы гарантировать целостность данных. Например, если запись нарушает установленное ограничение, система выдает предупреждение. При удалении записей также проверяются зависимости, чтобы избежать потери связанных данных.

Также модуль обеспечивает импорт данных из CSV или XLSX файлов, что значительно упрощает процесс наполнения таблиц информацией. Пользователь может загрузить готовые данные из внешних источников, таких как электронные таблицы или выгрузки, и автоматически преобразовать их в структуру таблицы. При импорте система проверяет соответствие типов данных и структуры столбцов, чтобы гарантировать корректность загружаемой информации.

#### 3.12. Модуль управления вычислителями

Модуль предназначен для управления вычислителями, включая их создание, редактирование, удаление, а также хранение и получение данных. При создании или редактировании вычислителя пользователь может задать его название и описание, а также загрузить ZIP-архив со скриптом.

Архив должен содержать:

- о исполняемый файл main.py с логикой выполнения скрипта,
- файл requirements.txt со списком зависимостей для установки через pip.

Для корректной работы скрипт должен быть совместим с актуальными версиями Python.

#### 3.13. Модуль управления формами параметров

Модуль управления формами предназначен для создания, редактирования и удаления форм, а также для хранения и получения данных

о них. Формы выступают в роли шаблонов, которые определяют набор параметров, необходимых для выполнения расчетов. Каждая форма может быть настроена под конкретные задачи, что делает процесс подготовки входных данных удобным и стандартизированным.

При создании или редактировании формы пользователь может задать её название и настроить параметры. Параметры представляют собой ключевые элементы формы, которые определяют структуру входных данных. Для каждого параметра можно указать:

- о название,
- о тип данных,
- о ключ,
- значение по умолчанию
- обязательность заполнения.

Также предусмотрена возможность задания ограничений, которые будут проверяться при вводе данных. Например, можно указать, что параметр должен быть больше нуля. Ограничения создаются путем указания левого операнда (параметр), оператора (например, "больше", "меньше" или "равно") и правого операнда (значение).

Важной особенностью каждой формы является наличие дефолтных параметров, которые являются обязательными для заполнения. Эти параметры присутствуют в каждой форме и не могут быть удалены. К таким параметрам относятся:

- Вычислитель (указывает, какой вычислитель будет использоваться для обработки данных),
- Таблицы (определяет таблицы, из которых будут браться данные для расчетов),
- Время моделирования (задает временной интервал в секундах, в течение которого будет выполняться моделирование).

• Дашборд (определяет дашборд, который будет использоваться для визуализации результатов расчетов).

После создания формы её можно использовать для подготовки входных данных. При создании расчета пользователь указывает, из какой формы следует брать параметры. На основе выбранной формы система автоматически генерирует интерфейс для ввода значений, проверяя их на соответствие заданным характеристикам. Это гарантирует, что входные данные будут корректными и соответствовать требованиям расчетов.

Модуль управления формами предлагает ряд преимуществ. Он стандартизирует входные данные, что упрощает их использование в расчетах и минимизирует риск ошибок. Гибкость настройки позволяет адаптировать параметры под конкретные задачи, а проверка ограничений и обязательности заполнения обеспечивает целостность данных.

#### 3.14. Модуль управления расчетами

Модуль управления расчетами предназначен для создания, редактирования, удаления расчетов, а также для их запуска и остановки. Статус расчета динамически изменяется в зависимости от прогресса выполнения задачи и действий пользователей, влияющих на процесс. Это позволяет точно отслеживать текущее состояние расчета на каждом этапе его жизненного цикла.

После создания расчет автоматически получает статус "Создан", что подтверждает его успешную инициализацию. По мере продвижения расчет проходит через различные статусы, которые отражают его текущую фазу выполнения:

Подготовка входных данных – система собирает и настраивает данные, необходимые для выполнения расчета.

- Готов к запуску все данные подготовлены, и расчет ожидает начала выполнения,
- В очереди расчет находится в списке задач, ожидающих обработки системой,
- Ожидает скачивания выполняется подготовка к загрузке необходимых файлов или данных,
- Скачивание данные активно загружаются для дальнейшего использования в расчетах,
- В работе расчет выполняется, и система обрабатывает заданные параметры.

Финальные статусы расчета определяются результатом выполнения задачи или действиями пользователя:

- Успешно расчет завершен без ошибок, и результаты доступны для анализа,
- Ошибка во время выполнения произошел сбой или возникли проблемы с данными,
- Отменен сервером расчет был принудительно остановлен сервером по техническим или административным причинам,
- Отменен хостом пользователь или хост инициировал остановку расчета,
- Время истекло расчет был автоматически завершен из-за превышения заданного времени моделирования,
- Архивирован расчет был удален пользователем.

Статус расчета автоматически обновляется в реальном времени в зависимости от действий пользователя (например, запуск, остановка или отмена) или событий в системе (завершение обработки, ошибка или истечение времени).

#### 3.15. Модуль управления дашбордами

Модуль управления дашбордами предоставляет пользователям возможность создавать, редактировать, удалять и использовать шаблоны для визуализации данных расчетов. Он позволяет организовать процесс работы с дашбордами, связывая их с вычислителями и расчетами.

При создании нового шаблона пользователь задает его название, выбирает вычислитель, к которому он будет привязан, и указывает, является ли он шаблоном по умолчанию для данного вычислителя. После создания шаблон автоматически связывается с выбранным вычислителем. Если шаблон больше не нужен, модуль позволяет его удалить.

Редактирование шаблонов позволяет изменять название, добавлять или удалять виджеты, а также настраивать их параметры. Пользователь может изменять расположение виджетов с помощью функции drag-and-drop, а также корректировать их размеры и пропорции. Для каждого типа виджета доступны специфические настройки, такие как выбор источника данных, настройка осей графиков, цветовая палитра и другие параметры. Если возникла ошибка, пользователь может отменить последние действия (undo) и вернуться к предыдущей версии шаблона.

Шаблоны могут быть указаны в качестве параметров расчетов или форм. Выбор шаблона возможен только после выбора связанного вычислителя.

Просмотр результатов расчета доступен только для успешно завершенных расчетов. При применении шаблона система автоматически заполняет виджеты данными из результатов расчета. В режиме просмотра пользователь может взаимодействовать с виджетами, изменяя масштаб графиков, применяя фильтры и настраивая параметры отображения. Однако редактирование структуры дашборда в этом режиме недоступно – он работает только в режиме "только для чтения". Модуль управления шаблонами дашбордов обеспечивает удобное управление визуализацией данных, позволяя пользователям легко создавать, настраивать и использовать шаблоны для анализа и представления информации. Благодаря поддержке различных типов виджетов и гибким настройкам, модуль становится универсальным инструментом для работы с данными любой сложности.

#### 3.16. Модуль управления приложением

Модуль управления приложением отвечает за запуск работы приложения и настройку его ключевых параметров. Он позволяет конфигурировать режим работы, базы данных, сервисы аутентификации, параметры безопасности и другие системные настройки, обеспечивая гибкость и управление основными аспектами системы.

Кроме того, модуль управляет дизайном приложения и настройками внешних ссылок. Он предоставляет возможность добавлять, редактировать и удалять ссылки, а также загружать логотипы для страницы авторизации и самого приложения. Модуль также позволяет изменять цветовую палитру приложения, чтобы настроить его внешний вид в соответствии с требованиями.

Модуль обеспечивает интеграцию как с другими модулями приложения, так и с внешними сервисами, что позволяет расширять функциональность системы и взаимодействовать с внешними источниками данных.

Кроме того, модуль управляет процессом обработки и запуска расчетов. Это включает в себя предоставление доступных для вычислений задач сервису обработки, выгрузку данных задачи, синхронизацию статуса расчета и выгрузку результатов расчетов обратно в систему.

#### 3.17. Вспомогательные модули

В данном разделе представлены модули, обеспечивающие общие конфигурации, настройки сборки и локализацию системы.

- Модуль etl предоставляет базовые инструменты для импорта и экспорта данных, которые задействуются другими модулями.
- Модуль common содержит базовые конфигурации и классы, используемые в других модулях.
- Модуль buildSrc вспомогательный Gradle-проект с необходимыми для сборки скриптами, конфигурациями и плагинами.
- Модуль gradle директория для хранения спецификаций версий зависимостей проекта и конфигурации сервиса обработки вычислений.
- Модуль i18n отвечает за переводы системных сообщений.

Эти модули обеспечивают стабильную работу системы, упрощая управление зависимостями, конфигурациями и локализацией.

#### 3.18. Модуль пользовательского интерфейса

Модуль предоставляет интуитивно понятную систему навигации по основным разделам и функциям приложения. Пользователи могут быстро находить ключевые элементы системы благодаря четко организованной структуре интерфейса.

В модуле используются стандартные элементы интерфейса. Применение привычных компонентов уменьшает время на освоение системы и делает работу пользователя более комфортной.

Система обеспечивает мгновенную обратную связь при каждом действии пользователя. Например, при нажатии кнопки или отправке данных пользователь сразу видит результат своих действий.

Модуль активно уведомляет пользователей об успешных действиях. Это может быть визуальное подтверждение, например сообщение об

успешной отправке данных или выполнении операции. Такая прозрачность работы системы позволяет пользователям быть уверенными в том, что их действия были корректно обработаны

Поддерживается модульная структура интерфейса, где каждый компонент работает независимо от других. Это позволяет легко изменять, добавлять или удалять отдельные элементы без необходимости перерабатывать всю систему. Модульность делает интерфейс гибким и адаптируемым под различные задачи.

Кроме того, модуль предоставляет возможность переиспользования интерфейсных компонентов в разных частях системы. Это не только упрощает процесс разработки, но и обеспечивает целостность стиля и поведения элементов интерфейса.

### 3.19. Сервис обработки задач

Сервис обработки задач представляет собой распределенную систему, состоящую из серверной части и клиентской части. Система предназначена для автоматизации процессов поиска, выполнения и управления вычислительными задачами.

Серверная часть является основным компонентом системы, отвечающим за выполнение задач. Она работает автономно и может быть развернута в нескольких экземплярах на различных серверах.

Основные функции серверной части:

- 1. Поиск задач автоматический поиск доступных задач для выполнения.
- 2. Скачивание задач загрузка данных задач из веб-приложения.
- 3. Выполнение задач обработка задач.
- 4. Выгрузка результатов отправка результатов выполнения в вебприложение.

5. Оповещение о прогрессе выполнения задачи

Особенности работы серверной части:

• Возможность запуска нескольких экземпляров серверов в различных локациях.

- Автономная работа без необходимости постоянного взаимодействия с клиентской частью.
- Возможность добавления нескольких исполнителей.

Клиентская часть представляет собой удобный инструмент для управления и мониторинга работы серверов. Она предоставляет пользователю полный контроль над всеми процессами, происходящими на серверах, а также возможность настройки и оптимизации их работы.

Основные функции клиентской части:

- Подключение к нескольким серверам Пользователь может подключаться к любому количеству серверов, независимо от их расположения. Это позволяет централизованно управлять всеми серверами через единый интерфейс.
- 2. Мониторинг вычислителей и задач Клиентское приложение предоставляет информацию о текущем состоянии вычислителей и задач, выполняемых на серверах. Пользователь может видеть прогресс выполнения задач, их статус и другую информацию.
- 3. Добавление и настройка вычислителей Через клиентское приложение можно добавлять новые вычислители на серверы и настраивать их параметры.
- 4. Остановка
   выполняемых
   задач

   При необходимости пользователь может остановить выполнение
   задачи на сервере.
- Завершение работы сервера Через клиентское приложение можно безопасно завершить работу сервера, убедившись, что все текущие задачи завершены или корректно остановлены.

# 4. Пользовательские сценарии

## Самостоятельная регистрация пользователя

Краткое описание	Самостоятельная регистрация пользователя в
сценария	системе
Участники	Пользователь
	Администратор
	Система
Предусловие	Пользователь не зарегистрирован в системе.
(контекст)	Система предоставляет возможность для
	самостоятельной регистрации.
	Администратор может настроить формат адреса
	электронной почты.
Постусловие	Пользователь зарегистрирован в системе.
(результат)	
Основной сценарий	1. Пользователь заходит на главную страницу.
	2. Пользователь нажимает кнопку
	"Зарегистрироваться".
	3. Пользователь вводит имя, фамилию и e-mail и
	нажимает кнопку "Создать аккаунт".
	4. Система проверяет наличие шаблона e-mail,
	установленного администратором:
	4а. Если шаблон не установлен или e-mail
	соответствует шаблону, то переходим к п.6.
	4б. Если e-mail не соответствует шаблону, то
	регистрация требует подтверждения
	администратора, переходим к п.5.
	5. Администратор подтверждает регистрацию.

	6. Система отправляет письмо со ссылкой для
	активации на почту пользователя.
	7. Пользователь переходит по ссылке из письма и
	перенаправляется на страницу для создания пароля.
	8. Пользователь вводит новый пароль, подтверждает
	его и нажимает кнопку "Создать".
	9. Система завершает процесс регистрации и
	активирует аккаунт пользователя.
	10. Пользователь получает возможность входа в
	систему.
Альтернативный	A1. Неверный формат e-mail
сценарий	1. Система отображает сообщение об ошибке и
	просит пользователя ввести корректный e-mail.
	A2. Пользователь с указанным e-mail уже
	существует
	1. Система отображает сообщение об ошибке и
	просит пользователя ввести другой e-mail.
	АЗ. Администратор отклоняет регистрацию
	пользователя
	1. Система не отправляет пользователю код
	активации.
	А4. Пароль пользователя не отвечает требованиям
	политики
	1. Система отображает сообщение об ошибке,
	указывая конкретный критерий безопасности,
	которому не соответствует пароль.
Исключения	Сбой в работе системы, отсутствие доступа к панели
	администратора и/или почтовому серверу.

Краткое описание	Регистрация пользователя через администратора
сценария	
Участники	Пользователь
	Администратор
	Система
Предусловие	1. Администратор авторизован в системе.
(контекст)	2. Пользователь не зарегистрирован в системе.
Постусловие	Пользователь зарегистрирован в системе.
(результат)	
Основной сценарий	1. Администратор переходит в раздел
	"Администрирование" на вкладку "Пользователи".
	2. Администратор нажимает кнопку "Создать
	аккаунт".
	3. Администратор вводит имя, фамилию, e-mail
	пользователя и нажимает кнопку "Создать аккаунт".
	4. Система отправляет письмо со ссылкой для
	активации на почту пользователя.
	5. Пользователь переходит по ссылке из письма и
	перенаправляется на страницу для создания пароля.
	6. Пользователь вводит новый пароль, подтверждает
	его и нажимает кнопку "Создать".
	7. Система завершает процесс регистрации и
	активирует аккаунт пользователя.
	8. Пользователь получает возможность входа в
	систему.
Альтернативный	A1. Неверный формат e-mail
сценарий	

# Регистрация пользователя через администратора

	1. Система отображает сообщение об ошибке и
	просит администратора ввести корректный e-mail.
	A2. Пользователь с указанным e-mail уже
	существует
	1. Система отображает сообщение об ошибке и
	просит администратора ввести другой e-mail.
	А4. Пароль пользователя не отвечает требованиям
	политики
	1. Система отображает сообщение об ошибке,
	указывая конкретный критерий безопасности,
	которому не соответствует пароль.
Исключения	Сбой в работе системы, отсутствие доступа к панели
	администратора и/или почтовому серверу.

# Вход пользователя в систему через внутреннюю учетную запись

Краткое описание	Вход пользователя в систему через внутреннюю
сценария	учетную запись
Участники	Пользователь
	Система
Предусловие	Пользователь зарегистрирован в системе.
(контекст)	
Постусловие	Пользователь получил доступ к системе.
(результат)	
Основной	1. Пользователь заходит на главную страницу.
сценарий	2. Пользователь вводит логин, пароль, (при
	необходимости активирует флажок "Запомнить
	меня") и нажимает кнопку "Войти".

	3. Система идентифицирует пользователя, проверяя,
	имеется ли в локальной базе данных пользователь с
	указанным логином.
	4. Система аутентифицирует пользователя, проверяя
	правильность введенного пароля.
	5. Система авторизует пользователя, предоставляя
	ему доступ к функционалу и ресурсам,
	соответствующим его роли и правам, установленным
	в системе.
	6. Система отображает вкладку "Проекты" с деревом
	групп и проектов (если они были созданы ранее).
Альтернативный	А1. Пользователя с указанным логином не
сценарий	существует
	1. Система отображает сообщение об ошибке и
	просит пользователя ввести корректные данные.
	2. Пользователь исправляет ошибку и повторяет
	попытку входа.
	А2. Указанный пользователем пароль не совпадает
	1. Система отображает сообщение об ошибке и
	просит пользователя ввести корректные данные.
	2. Пользователь исправляет ошибку и повторяет
	попытку входа.
	АЗ. Пользователь заблокирован
	1. Система отображает сообщение о блокировке
	пользователя.
	А4. Истек срок действия пароля
	1. Система отображает сообщение об ошибке и
	рекомендует сменить пароль.

Исключения	Сбой в работе системы.

# Вход пользователя в систему через LDAP

Краткое описание	Вход пользователя в систему через LDAP
сценария	
Участники	Пользователь
	Система
	Служба каталогов (OpenLDAP или Active Directory)
Предусловие	Пользователь зарегистрирован в службе каталогов.
(контекст)	
Постусловие	Пользователь получил доступ к системе.
(результат)	
Основной сценарий	1. Пользователь заходит на главную страницу.
	2. Пользователь вводит логин, пароль, и нажимает
	на кнопку "Войти".
	3. Система отправляет полученные данные на
	проверку в службу каталогов.
	4. Служба каталогов проверяет учетные данные.
	5. Служба каталогов возвращает информацию о
	пользователе.
	6. Система соотносит роли пользователя,
	полученные из службы каталогов, с внутренними
	правами доступа.
	7. Система авторизует пользователя, предоставляя
	ему доступ к функционалу и ресурсам,
	соответствующим его правам.

	8. Система отображает вкладку "Проекты" с
	деревом групп и проектов (если они были созданы
	ранее).
Альтернативный	А1. Служба каталогов вернула ошибку
сценарий	аутентификации
	1. Система отображает сообщение об ошибке.
	А2. Пользователь заблокирован в службе каталогов
	1. Система отображает сообщение о блокировке
	пользователя.
	АЗ. Истек срок действия пароля в службе
	каталогов
	1. Система отображает сообщение об ошибке и
	рекомендует сменить пароль.
Исключения	Сбой в работе системы, отсутствие доступа к
	службе каталога.

# Создание проекта

Краткое описание	Предоставить пользователю возможность создавать
сценария	проекты
Участники	Пользователь
	Система
Предусловие	Пользователь авторизован в системе.
(контекст)	Пользователь находится на вкладке «Проекты».
Постусловие	Пользователь успешно создал проект.
(результат)	
Основной	1. Пользователь нажимает на кнопку «+» в левом
сценарий	верхнем углу.

2. В выпадающем списке пользователь нажимает на
кнопку «Проект».
3. В модальном окне пользователь указывает
название и описание проекта, а также выбирает
родительскую группу или корневой уровень, к
которому будет относиться создаваемый проект.
4. После заполнения всех необходимых полей
пользователь нажимает кнопку "Сохранить".
5. Система создает проект на основе введенных
данных.
6. Система автоматически добавляет создателя
проекта в список участников, присваивая ему
дефолтную роль Owner.
7. Система отображает пользователю сообщение
«Успешно создано».
8. Система отображает проект в дереве на уровне
вложенности, заданном при создании, с указанным
названием.
9. Пользователь нажимает на название проекта.
10. Система отображает пользователю страницу
проекта, включая:
10.1. Название проекта.
10.2. Описание проекта.
10.3. Панель инструментов для управление
проектом (изменение метаданных, экспорт и
удаление проекта).
10.3. Панель навигации с доступными
разделами (Таблицы, Вычислители, Формы,

	Расчеты, Дашборды, Настройки, Участники,
	События).
Альтернативный	4А. Пользователь не указал название проекта
сценарий	1. Система заблокирует возможность создания
	проекта.
	2. Система подсвечивает незаполненное поле
	красным цветом и отображает сообщение об ошибке
	«Обязательное поле» при наведении на это поле.
	4Б. Пользователь не выбирает родительскую группу
	или корневой уровень.
	1. Система автоматически присваивает проекту
	корневой уровень.
	4В. Пользователь отменяет создание проекта до
	сохранения:
	1. Система отменяет создание проекта и не сохраняет
	введенные данные.
Исключения	Сбой в работе системы.

# Создание таблицы

Краткое описание	Предоставить пользователю возможность создавать
сценария	таблицы в проекте
Участники	Пользователь
	Система
Предусловие	Пользователь авторизован в системе.
(контекст)	Пользователь создал проект.
	Пользователь находится в разделе «Таблицы».

Постусловие	Пользователь успешно создал таблицу.
(результат)	
Основной	1. Пользователь нажимает на кнопку «+» в левом
сценарий	верхнем углу.
	2. В выпадающем списке пользователь нажимает на
	кнопку «Таблица».
	3. В модальном окне пользователь указывает
	название таблицы, а также выбирает родительскую
	директорию или корневой уровень, к которому будет
	относиться создаваемая таблица.
	4. После заполнения всех необходимых полей
	пользователь нажимает кнопку "Сохранить".
	5. Система создает таблицу на основе введенных
	данных.
	6. Система отображает пользователю сообщение
	«Успешно создано».
	7. Система отображает таблицу в дереве на уровне
	вложенности, заданном при создании, с указанным
	названием.
	8. Система направляет пользователя в раздел
	«Структура» для задания структуры созданной
	таблицы.
Альтернативный	4А. Пользователь не указал название таблицы
сценарий	1. Система заблокирует возможность создания
	таблицы.
	2. Система подсвечивает незаполненное поле
	красным цветом и отображает сообщение об ошибке
	«Обязательное поле» при наведении на это поле.

	4Б. Пользователь указывает название таблицы,
	которая уже существует на выбранном уровне
	дерева
	1. Система заблокирует возможность создания
	таблицы.
	2. Система подсвечивает поле «Название» красным
	цветом и отображает сообщение об ошибке «Объект
	с таким именем уже существует на этом уровне» при
	наведении на это поле.
	4В. Пользователь не выбирает родительскую
	директорию или корневой уровень.
	1. Система автоматически привязывает таблицу к
	корневому уровню.
	4Г. Пользователь отменяет создание таблицы до
	сохранения:
	1. Система отменяет создание таблицы и не
	сохраняет введенные данные.
Исключения	Сбой в работе системы.

# Настройка структуры таблицы

Краткое описание	Предоставить пользователю возможность задавать
сценария	структуру таблицы
Участники	Пользователь
	Система

Предусловие	Пользователь авторизован в системе.
(контекст)	Пользователь создал проект.
	Пользователь находится на вкладке «Таблицы».
	Пользователь создал таблицу.
	Пользователь находится в разделе «Структура»
	созданной таблицы.
Постусловие	Пользователь успешно задал структуру таблицы.
(результат)	
Основной	1. Система отображает пользователю перечень
сценарий	столбцов с описанием их характеристик (название,
	тип данных, и т.д.).
	2. Пользователь нажимает на кнопку «Добавить
	столбец таблицы» в шапке таблицы.
	3. Система отображает пользователю модальное окно
	создания столбца таблицы.
	4. Пользователь задает:
	4.1. Название столбца
	4.2. Тип данных столбца
	4.3. Признак уникальности столбца
	4.4. Признак обязательности столбца
	5. После заполнения всех необходимых полей
	пользователь нажимает кнопку "Сохранить".
	6. Система создает столбец на основе введенных
	данных.
	7. Система отображает пользователю сообщение
	«Успешно создано».
	8. Созданный столбец отображается в перечне
	столбцов.

Альтернативный	5А. Пользователь не указал название столбца
сценарий	1. Система заблокирует возможность создания
	столбца.
	2. Система подсвечивает незаполненное поле
	красным цветом и отображает сообщение об ошибке
	«Обязательное поле» при наведении на это поле.
	5Б. Пользователь указывает название столбца,
	который уже существует в таблице
	1. Система заблокирует возможность создания
	столбца.
	2. Система подсвечивает поле «Название» красным
	цветом и отображает сообщение об ошибке «В
	таблице уже существует столбец с таким названием»
	при наведении на это поле.
	5В. Пользователь отменяет создание столбца до
	сохранения:
	1. Система отменяет создание столбца и не сохраняет
	введенные данные.
Исключения	Сбой в работе системы.

# Наложение ограничений на столбцы таблицы

Краткое описание	Предоставить пользователю возможность
сценария	накладывать ограничения на столбцы таблицы
Участники	Пользователь
	Система

Предусловие	Пользователь авторизован в системе.
(контекст)	Пользователь создал проект.
	Пользователь находится на вкладке «Таблицы».
	Пользователь создал таблицу.
	Пользователь создал хотя бы один столбец.
Постусловие	Пользователь успешно создал ограничение на
(результат)	столбец таблицы.
Основной сценарий	1. Пользователь переходит на вкладку
	«Ограничения».
	2. Система отображает пользователю перечень
	ограничений с описанием их характеристик (левая
	часть оператора, операнд, правая часть оператора).
	3. Пользователь нажимает на кнопку «Добавить
	ограничение» в шапке таблицы.
	4. Система отображает пользователю модальное
	окно создания ограничения.
	5. Пользователь задает:
	5.1. Левую часть оператора
	5.2. Операнд
	5.3. Правую часть оператора
	6. После заполнения всех необходимых полей
	пользователь нажимает кнопку "Сохранить".
	7. Система создает ограничение на основе
	введенных данных.
	7. Система отображает пользователю сообщение
	«Успешно создано».
	8. Созданное ограничение отображается в перечне
	ограничений.

Альтернативный	6А. Пользователь не указал левую/правую части
сценарий	оператора
	1. Система заблокирует возможность создания
	ограничения.
	2. Система подсвечивает незаполненное поле
	красным цветом и отображает сообщение об ошибке
	«Обязательное поле» при наведении на это поле.
	6Б. Ограничение, заданное пользователем,
	противоречит данным в таблице
	1. Система заблокирует возможность создания
	ограничения.
	2. Система отображает сообщение об ошибке «Не
	удалось создать ограничение в таблице».
	6В. Пользователь отменяет создание ограничения
	до сохранения:
	1. Система отменяет создание ограничения и не
	сохраняет введенные данные.
Исключения	Сбой в работе системы.

# Добавление данных в таблицу

Краткое описание	Предоставить пользователю возможность добавлять
сценария	данные в таблицу
Участники	Пользователь
	Система
Предусловие	Пользователь авторизован в системе.
(контекст)	Пользователь создал проект.

	Пользователь находится на вкладке «Таблицы».
	Пользователь создал таблицу.
	Пользователь задал структуру таблицы (создал хотя
	бы один столбец и, если необходимо, наложил
	ограничения).
Постусловие	Пользователь успешно добавил данные в таблицу.
(результат)	
Основной	1. Пользователь переходит в раздел «Данные».
сценарий	2. Система отображает пользователю таблицу с
	заданными столбцами.
	3. Пользователь нажимает на кнопку «Добавить
	данные» в шапке таблицы.
	4. Система отображает пользователю модальное окно
	добавления данных, где отображается название ранее
	созданного столбца и предоставляется поле для
	ввода значения.
	5. Пользователь заполняет поле и нажимает кнопку
	"Сохранить".
	6. Система создает запись на основе введенных
	данных.
	7. Система отображает пользователю сообщение
	«Успешно создано».
	8. Созданная запись отображается в таблице.
Альтернативный	5А. Пользователь не заполнил обязательные поля.
сценарий	1. Система заблокирует возможность добавления
	данных.

	2. Система подсвечивает незаполненное поле
	красным цветом и отображает сообщение об ошибке
	«Обязательное поле» при наведении на это поле.
	5Б. Введенное значение не соответствует
	ограничению
	1. Система заблокирует возможность добавления
	данных.
	2. Система подсвечивает неправильно заполненное
	поле красным цветом и отображает сообщение о
	несоответствии ограничению при наведении на это
	поле.
	5В. Пользователь отменяет добавление данных
	1. Система отменяет создание записи в таблице и не
	сохраняет введенные данные.
Исключения	Сбой в работе системы.

# Создание вычислителя

Краткое описание	Предоставить пользователю возможность добавлять
сценария	данные в таблицу
Участники	Пользователь
	Система
Предусловие	Пользователь авторизован в системе.
(контекст)	Пользователь создал проект.
	Пользователь создал и заполнил данными таблицу.
Постусловие	Пользователь успешно создал вычислитель.
(результат)	

Основной	1. Пользователь переходит на вкладку
сценарий	«Вычислители».
	2. Система отображает таблицу с перечнем
	вычислителей.
	3. Пользователь нажимает на кнопку «Добавить» в
	шапке таблицы.
	4. Система отображает пользователю модальное окно
	добавления вычислителя.
	5. Пользователь задает название и описание
	вычислителя, а также загружает ZIP-архив со
	скриптом.
	6. Пользователь нажимает кнопку "Сохранить".
	7. Система создает вычислитель.
	8. Система отображает пользователю сообщение
	«Успешно создано».
	9. Созданный вычислитель отображается в таблице.
Альтернативный	6А. Пользователь не заполнил обязательные поля
сценарий	1. Система заблокирует возможность создание
	вычислителя.
	2. Система подсвечивает незаполненное поле
	красным цветом и отображает сообщение об ошибке
	«Обязательное поле» при наведении на это поле.
	6Б. Пользователь указывает название вычислителя,
	который уже существует в системе
	1. Система заблокирует возможность создания
	вычислителя.

	2. Система подсвечивает поле «Название» красным
	цветом и отображает сообщение об ошибке «Поле не
	должно содержать следующие значения:» при
	наведении на это поле.
	6В. Пользователь загружает файл неправильного
	формата
	1. Система заблокирует возможность создания
	вычислителя.
	2. Система отображает сообщение об ошибке
	«Неподдерживаемое расширение для загрузки
	файла:».
	6Г. Пользователь загружает файл, размер которого
	превышает допустимый лимит
	1. Система создает вычислитель, не прикрепляя к
	нему невалидный ZIP-архив.
	2. Система отображает сообщение «Превышен
	максимальный размер файла».
	6Д. Пользователь отменяет создание вычислителя
	1. Система отменяет создание вычислителя и не
	сохраняет введенные данные.
Исключения	Сбой в работе системы.

# Создание формы параметров

Краткое описание	Предоставить пользователю возможность создавать
сценария	форму параметров.

Участники	Пользователь
	Система
Предусловие	Пользователь авторизован в системе.
(контекст)	Пользователь создал проект.
	Пользователь создал и заполнил данными таблицу.
	Пользователь создал вычислитель.
	Пользователь находится на странице проекта.
Постусловие	Пользователь успешно создал форму параметров.
(результат)	
Основной	1. Пользователь переходит на вкладку «Формы».
сценарий	2. Система отображает пустую таблицу.
	3. Пользователь нажимает на кнопку «Добавить» в
	шапке таблицы.
	4. Система отображает пользователю модальное окно
	создания формы.
	5. Пользователь задает название формы и нажимает
	кнопку «Сохранить».
	6. Система выводит страницу редактирования формы,
	содержащую таблицу с параметрами. В этой таблице
	отображаются три обязательных параметра, общих
	для всех форм: «Вычислитель», «Время
	моделирования, с», «Таблицы» и «Дашборд».
	7. [опционально] Пользователь нажимает кнопку
	редактирования дефолтного параметра.
	8. Система отображает модальное окно
	редактирования параметра.

9. Пользователь задает значение по умолчанию для
дефолтного параметра и нажимает кнопку
«Сохранить».
10. Система сохраняет изменения и отображает
сообщение «Успешно обновлено».
11. Система отображает изменения в таблице
параметров.
12. Пользователь нажимает кнопку «Добавить» в
шапке таблицы.
13. Система отображает модальное окно добавления
параметра.
14. Пользователь задаёт:
14.1. Название параметра.
14.2. Тип параметра.
14.3. Ключ параметра.
14.4. Значение по умолчанию.
14.5. Обязательность заполнения параметра.
15. Пользователь нажимает кнопку «Сохранить».
16. Система создает параметр формы.
17. Система отображает сообщение «Успешно
создано»
18. Созданный параметр отображается в таблице
параметров.
19. [опционально] Пользователь нажимает кнопку
«Редактировать ограничения» в контекстном меню
параметра.
20. Система отображает таблицу ограничений справа
от таблицы параметров.

	21. Пользователь нажимает кнопку «Добавить
	ограничение» в шапке таблицы.
	22. Система отображает модальное окно наложения
	ограничения на параметр.
	23. Пользователь задает оператор, правый операнд и
	нажимает кнопку «Сохранить».
	24. Система отображает созданное ограничение в
	соответствующей таблице.
	25. Пользователь нажимает кнопку «Сохранить».
	26. Система отображает пользователю сообщение
	«Успешно обновлено».
	27. Система отображает страницу с таблицей форм,
	содержащей также форму, созданную пользователем
	ранее.
Альтернативный	9А. Пользователь указал невалидное значение по
сценарий	умолчанию для параметра «Время моделирования, с».
сценарий	умолчанию для параметра «Время моделирования, с». 1. Система заблокирует возможность редактирования
сценарий	умолчанию для параметра «Время моделирования, с». 1. Система заблокирует возможность редактирования параметра.
сценарий	<ul> <li>умолчанию для параметра «Время моделирования, с».</li> <li>1. Система заблокирует возможность редактирования параметра.</li> <li>2. Система подсвечивает поле красным цветом и</li> </ul>
сценарий	<ul> <li>умолчанию для параметра «Время моделирования, с».</li> <li>1. Система заблокирует возможность редактирования параметра.</li> <li>2. Система подсвечивает поле красным цветом и отображает сообщение об ошибке, например, «Число</li> </ul>
сценарий	<ul> <li>умолчанию для параметра «Время моделирования, с».</li> <li>1. Система заблокирует возможность редактирования параметра.</li> <li>2. Система подсвечивает поле красным цветом и отображает сообщение об ошибке, например, «Число должно быть больше или равно 0» при наведении на</li> </ul>
сценарий	<ul> <li>умолчанию для параметра «Время моделирования, с».</li> <li>1. Система заблокирует возможность редактирования параметра.</li> <li>2. Система подсвечивает поле красным цветом и отображает сообщение об ошибке, например, «Число должно быть больше или равно 0» при наведении на это поле.</li> </ul>
сценарий	<ul> <li>умолчанию для параметра «Время моделирования, с».</li> <li>1. Система заблокирует возможность редактирования параметра.</li> <li>2. Система подсвечивает поле красным цветом и отображает сообщение об ошибке, например, «Число должно быть больше или равно 0» при наведении на это поле.</li> </ul>
сценарий	умолчанию для параметра «Время моделирования, с». 1. Система заблокирует возможность редактирования параметра. 2. Система подсвечивает поле красным цветом и отображает сообщение об ошибке, например, «Число должно быть больше или равно 0» при наведении на это поле. 9Б. Пользователь отменяет создание
сценарий	умолчанию для параметра «Время моделирования, с». 1. Система заблокирует возможность редактирования параметра. 2. Система подсвечивает поле красным цветом и отображает сообщение об ошибке, например, «Число должно быть больше или равно 0» при наведении на это поле. 9Б. Пользователь отменяет создание редактирование параметра:
сценарий	<ul> <li>умолчанию для параметра «Время моделирования, с».</li> <li>1. Система заблокирует возможность редактирования параметра.</li> <li>2. Система подсвечивает поле красным цветом и отображает сообщение об ошибке, например, «Число должно быть больше или равно 0» при наведении на это поле.</li> <li>9Б. Пользователь отменяет создание редактирование параметра:</li> <li>1. Система отменяет редактирование параметра и не</li> </ul>
сценарий	умолчанию для параметра «Время моделирования, с». 1. Система заблокирует возможность редактирования параметра. 2. Система подсвечивает поле красным цветом и отображает сообщение об ошибке, например, «Число должно быть больше или равно 0» при наведении на это поле. 9Б. Пользователь отменяет создание редактирование параметра: 1. Система отменяет редактирование параметра и не сохраняет введенные данные.

15А. Пользователь не заполнил обязательные поля
1. Система заблокирует возможность создания
параметра.
2. Система подсвечивает незаполненные поля
красным цветом и отображает сообщение об ошибке
«Обязательное поле» при наведении на это поле.
15Б. Пользователь вводит некорректное значение
ключа
1. Система подсвечивает поле «Ключ» красным
цветом и отображает сообщение об ошибке «Поле
должно содержать только латинские буквы и цифры»
при наведении на это поле.
15В. Пользователь вводит некорректное значение по
умолчанию
1. Система подсвечивает поле «Значение по
умолчанию» красным цветом и отображает
сообщение об ошибке при наведении на это поле.
15Г. Пользователь указывает название параметра,
который уже существует в форме
1. Система заблокирует возможность создания
параметра и отобразит сообщение об ошибке.
15Д. Пользователь отменяет создание параметра
1. Система отменяет создание параметра и не
сохраняет введенные данные.

	23А. Пользователь не указал правую часть оператора
	1. Система заблокирует возможность создания
	ограничения.
	2. Система подсвечивает незаполненное поле красным
	цветом и отображает сообщение об ошибке
	«Обязательное поле» при наведении на это поле.
	23Б. Ограничение, заданное пользователем,
	противоречит значению по умолчанию параметра
	1. Система заблокирует возможность создания
	ограничения.
	2. Система отображает сообщение об ошибке
	«Значение не удовлетворяет ограничению».
	23В. Пользователь отменяет создание ограничения
	до сохранения:
	1. Система отменяет создание ограничения и не
	сохраняет введенные данные.
	254 How poor and a construction approximation to a construct the construction of the c
	25А. Пользователь отменяет созоание формы
	1. Система отменяет создание формы и не сохраняет
	введенные данные.
Исключения	Соой в работе системы.

# Создание и настройка дашборда

Краткое описание	Предоставить пользователю возможность создавать
сценария	дашборд

Участники	Пользователь
	Система
Предусловие	Пользователь авторизован в системе.
(контекст)	Пользователь создал проект.
	Пользователь создал вычислитель.
	Пользователь находится на странице проекта.
Постусловие	Пользователь успешно создал дашборд.
(результат)	
Основной	1. Пользователь переходит на вкладку «Дашборды».
сценарий	2. Система отображает пустую таблицу шаблонов
	дашбордов.
	3. Пользователь нажимает на кнопку «Добавить» в
	шапке таблицы.
	4. Система отображает модальное окно создания
	нового дашборда.
	5. Пользователь задает название дашборда, выбирает
	вычислитель, указывает, является ли дашборд
	дефолтным для данного вычислителя, и нажимает
	кнопку «Сохранить».
	6. Система отображает пустую область
	редактирования дашборда.
	7. Пользователь нажимает кнопку «Создать виджет».
	8. Система отображает выпадающий список с
	доступными типами виджетов (линейный график,
	круговая диаграмма, гистограмма, и т.д.).
	9. Пользователь выбирает тип виджета.
	10. Система отображает модальное окно
	редактирования виджета выбранного типа.

	11. Пользователь задает параметры виджета.
	12. Система отображает изменения в реальном
	времени в области предварительного просмотра
	виджета.
	13. Пользователь нажимает кнопку «Сохранить».
	14. Система сохраняет настройки виджета и
	отображает его в доступной области редактирования.
	15. [опционально] Пользователь меняет размер
	виджета и его позицию.
	16. Система сохраняет настройки виджета.
	17. Пользователь заканчивает настройку дашборда и
	нажимает кнопку «Сохранить».
	18. Система отображает страницу с таблицей
	шаблонов, содержащей также дашборд, созданный
	пользователем ранее.
Альтернативный	5А. Пользователь не заполнил обязательные поля
сценарий	1. Система заблокирует возможность созлания
	дашборда.
	дашборда. 2. Система подсвечивает незаполненные поля
	<ul><li>дашборда.</li><li>2. Система подсвечивает незаполненные поля</li><li>красным цветом и отображает сообщение об ошибке</li></ul>
	дашборда. 2. Система подсвечивает незаполненные поля красным цветом и отображает сообщение об ошибке «Обязательное поле» при наведении на это поле.
	дашборда. 2. Система подсвечивает незаполненные поля красным цветом и отображает сообщение об ошибке «Обязательное поле» при наведении на это поле.
	<ul> <li>дашборда.</li> <li>2. Система подсвечивает незаполненные поля красным цветом и отображает сообщение об ошибке «Обязательное поле» при наведении на это поле.</li> <li>5Б. Пользователь указывает название дашборда,</li> </ul>
	<ul> <li>дашборда.</li> <li>2. Система подсвечивает незаполненные поля красным цветом и отображает сообщение об ошибке «Обязательное поле» при наведении на это поле.</li> <li>5Б. Пользователь указывает название дашборда, который уже существует в системе</li> </ul>
	<ul> <li>дашборда.</li> <li>2. Система подсвечивает незаполненные поля красным цветом и отображает сообщение об ошибке «Обязательное поле» при наведении на это поле.</li> <li>5Б. Пользователь указывает название дашборда, который уже существует в системе</li> <li>1. Система заблокирует возможность создания</li> </ul>
	<ul> <li>дашборда.</li> <li>2. Система подсвечивает незаполненные поля красным цветом и отображает сообщение об ошибке «Обязательное поле» при наведении на это поле.</li> <li>5Б. Пользователь указывает название дашборда, который уже существует в системе</li> <li>1. Система заблокирует возможность создания дашборда.</li> </ul>

	2. Система отображает сообщение об ошибке
	«Дашборд с таким названием уже существует в
	системе».
	5В. Пользователь отменяет создание дашборда
	1. Система отменяет создание дашборда и не
	сохраняет введенные данные.
	11А. Пользователь не заполнил обязательные поля
	1. Система заблокирует возможность создания
	виджета.
	2. Система подсвечивает незаполненные поля
	красным цветом и отображает сообщение об ошибке
	«Обязательное поле» при наведении на это поле.
	17А. Пользователь отменяет создание дашборда
	1. Система отменяет создание дашборда и не
	сохраняет введенные данные.
Исключения	Сбой в работе системы.

# Интеграция с сервисом обработки задач

Краткое описание	Предоставить пользователю возможность
сценария	сгенерировать сервисный токен и подключить сервер
	для обработки вычислений
Участники	Пользователь
	Система
	Сервис обработки задач

Предусловие	Пользователь авторизован в системе.
(контекст)	Пользователь создал проект.
	Пользователь находится на странице проекта.
	Системный администратор предоставил
	пользователю токен для подключения сервера.
Постусловие	Пользователь успешно подключил сервис обработки
(результат)	вычислений к системе.
Основной	1. Пользователь переходит на вкладку «Настройки».
сценарий	2. Пользователь нажимает на раздел «Токены».
	3. Система отображает пустую таблицу токенов.
	4. Пользователь нажимает на кнопку «Сгенерировать
	токен» в шапке таблицы.
	4. Система отображает модальное окно генерации
	токена.
	5. Пользователь задает название токена, описание,
	дату истечения срока действия и роль.
	6. Пользователь нажимает кнопку «Сохранить».
	7. Система отображает пользователю
	сгенерированный токен.
	8. Пользователь нажимает кнопку «Скопировать».
	9. Система отображает сообщение «Токен скопирован
	в буфер обмена».
	10. Пользователь закрывает модальное окно.
	11. Система отображает страницу с таблицей токенов,
	содержащей также токен, созданный пользователем
	ранее.
	12. Пользователь открывает приложение для
	управления серверами обработки вычислений.

13. Сервис отображает пользователю список серверов
и обработчиков (если таковые есть).
13. Пользователь нажимает «+».
14. Сервис отображает окно для подключения к
серверу.
15. Пользователь вводит адрес сервера и токен,
переданный ему системным администратором.
16. Пользователь нажимает кнопку «Добавить».
17. Сервис добавляет новый сервер в список
доступных серверов и отображает его в перечне.
18. Пользователь нажимает на кнопку просмотра
информации о сервере ((i)).
19. Пользователь нажимает на кнопку «Добавить
исполнителя».
20. Пользователь задает:
20.1. Название
20.2. Токен (сгенерированный в п.7)
20.3. Рабочую директорию исполнителя
20.4. Путь к интерпретатору Python
20.5. Интервал поиска новых задач на расчет
20.6. Интервал оповещения о прогрессе задачи
20.7. Количество параллельно выполняемых
задач
21. Пользователь нажимает кнопку «Сохранить».
22. Сервис добавляет нового исполнителя и
отображает его в списке активных исполнителей для
выбранного сервера.

Альтернативный	5А. Дата истечения срока действия, указанная
сценарий	пользователем, меньше, чем текущая дата
	1. Система заблокирует возможность генерация
	токена.
	2. Система подсвечивает поле «Дата истечения срока
	действия» красным цветом и отображает сообщение
	об ошибке «Дата не может быть меньше <текущая
	дата>» при наведении на это поле.
	6А. Пользователь не заполнил обязательные поля
	1. Система заблокирует возможность генерации
	токена.
	2. Система подсвечивает незаполненные поля
	красным цветом и отображает сообщение об ошибке
	«Обязательное поле» при наведении на это поле.
	6Б. Пользователь отменил генерацию токена
	1. Система отменяет генерацию токена и не сохраняет
	введенные данные.
	8А. Пользователь закрыл модальное окно, не
	скопировав токен
	1. Система не предоставляет возможность повторного
	просмотра сгенерированного токена из соображений
	безопасности.
	2. Пользователь генерирует новый токен.
	16А. Сервер недоступен

	1. Сервис отображает пользователю сообщение об
	ошибке.
Исключения	Сбой в работе системы, сбой в работе сервиса
	обработки вычислений

# Создание и запуск расчета

Краткое описание	Предоставить пользователю возможность создать и
сценария	запустить расчет.
Участники	Пользователь
	Система
	Сервис обработки вычислений
Предусловие	Пользователь авторизован в системе.
(контекст)	Пользователь создал проект.
	Пользователь находится на странице проекта.
	Пользователь создал таблицу.
	Пользователь создал вычислитель.
	Пользователь создал форму.
Постусловие	Пользователь успешно запустил расчет.
(результат)	
Основной	1. Пользователь переходит на вкладку «Расчеты».
сценарий	2. Система отображает пустую таблицу расчетов.
	3. Пользователь нажимает кнопку «Создать расчет»
	в шапке таблицы.
	4. Система отображает модальное окно создания
	расчета.
	5. Пользователь задает название расчета, описание и
	форму параметров, которая определяет набор
	параметров, необходимых для выполнения расчета.

6. Система автоматически отображает параметры,
предусмотренные выбранной формой.
7. Пользователь вводит значения для каждого из
отображенных параметров.
8. Пользователь нажимает кнопку «Сохранить».
9. Система обновляет статус расчета на «Создан».
10. Начинается процесс подготовки данных для
расчета, о чем система информирует пользователя,
отображая статус «Подготовка входных данных».
11. После завершения подготовки данных система
автоматически переводит задачу в статус «Готов к
запуску» и отображает его пользователю.
12. Пользователь нажимает кнопку «Запустить
расчет».
13. Система отображает сообщение «Расчет успешно
запущен».
14. Система добавляет расчет в очередь и меняет
статус расчета с «Готов к запуску» на «В очереди»,
отображая его пользователю.
15. Сервис обработки задач периодически
обращается по токену в систему для получения
новых задач в статусе «В очереди».
16. Система оповещает Сервис обработки задач о
наличии подходящей задачи и меняет её статус на
«Ожидает скачивания», отображая его пользователю.
17. Сервис обработки задач запрашивает у системы
данные для расчета задачи.
17. Сервис обработки задач запрашивает у системы данные для расчета задачи.

	18. Система запаковывает необходимые данные в
	ZIP-архив и выгружает его сервису.
	19. Сервис обработки задач скачивает ZIP-архив,
	сообщая системе об этом.
	20. Система меняет статус задачи на «Скачивание»,
	отображая его пользователю.
	21. Сервис распаковывает ZIP-архив, настраивает
	среду выполнения, запускает задачу на расчет и
	оповещает об этом систему.
	22. Система меняет статус задачи на «В работе» и
	отображает его пользователю.
	23. После окончания расчета сервис обработки задач
	выгружает результаты задачи в виде ZIP-архива в
	систему.
	23. Система меняет статус задачи на «Успешно» и
	отображает его пользователю.
	24. [опционально] Пользователь нажимает кнопку
	скачивания результатов расчета в контекстном меню
	задачи.
Альтернативный	5А. Пользователь не заполнил обязательные поля
сценарий	1. Система заблокирует возможность создания
	расчета.
	2. Система подсвечивает незаполненные поля
	красным цветом и отображает сообщение об ошибке
	«Обязательное поле» при наведении на это поле.
	5Б. Пользователь отменил создание расчета

1. Система отменяет создание расчета и не сохраняет
введенные данные.
8А. Пользователь не заполнил обязательные поля
1. Система заблокирует возможность создания
расчета.
2. Система подсвечивает незаполненные поля
красным цветом и отображает сообщение об ошибке
«Обязательное поле» при наведении на это поле.
8Б. Пользователь отменил создание расчета
1. Система отменяет создание расчета и не сохраняет
введенные данные.
15А. Срок действия токена истек
1. Система отказывает Сервису обработки задач в
доступе в систему.
15Б. Сервис обработки задач имеет недостаточно
прав на получение данных задачи
1. Система отказывает Сервису обработки задач в
доступе в систему.
17А. Сервис обработки задач имеет недостаточно
прав на получение данных задачи
1. Система отказывает Сервису обработки задач в
доступе в систему.

	19А. Сервис обработки задач имеет недостаточно
	прав на получение данных задачи
	1. Система отказывает Сервису обработки задач в
	доступе в систему.
	21А. Структура ZIP-архива не соответствует
	требованиям
	1. Сервис обработки задач оповещает об этом
	систему.
	2. Система меняет статус задачи на «Ошибка».
	21Б. Возникли проблемы с настройкой среды
	выполнения
	1. Сервис обработки задач оповещает об этом
	систему.
	2. Система меняет статус задачи на «Ошибка».
Исключения	Сбой в работе системы, сбой в работе сервиса
	обработки вычислений

# Просмотр результатов расчета на дашборде

Краткое описание	Предоставить пользователю возможность
сценария	просматривать результаты расчета на дашборде
Участники	Пользователь
	Система
Предусловие	Пользователь авторизован в системе.
(контекст)	Пользователь находится на странице проекта.
	Пользователь создал и запустил расчет.
	Расчет завершен с результатом «Успешно».

	Пользователь находится на вкладке «Расчеты».
Постусловие	Пользователь успешно просмотрел результаты
(результат)	расчета на дашборде.
Основной сценарий	1. Пользователь нажимает кнопку просмотра
	результатов расчета.
	2. Система проверяет, какой шаблон дашборда
	указан в параметрах задачи.
	3. Система перенаправляет пользователя на новую
	вкладку и отображает дашборд, заполняя виджеты
	данными из результатов расчета.
Альтернативный	2А. Пользователь не указал дашборд в параметрах
сценарий	задачи
	1. Система определяет, какой вычислитель был
	применен для расчета задачи.
	2. Система проверяет, задан ли шаблон по
	умолчанию для вычислителя.
	3.а. Если шаблон задан, система отображает
	результаты расчета с использованием этого шаблона
	дашборда.
	3.б. Если шаблон не задан, система предлагает
	пользователю выбрать шаблона дашборда для
	отображения результатов.
	3А. Система не смогла применить шаблон
	дашборда к результатам расчета
	1. Система выделяет красным цветом виджеты,
	которые не удалось сгенерировать.
Исключения	Сбой в работе системы