

Документация, содержащая описание  
функциональных характеристик  
экземпляра программного обеспечения

1. Общие сведения .....	4
1.1. Назначение системы .....	4
1.2. Область применения.....	4
1.3. Функциональное назначение.....	4
1.4. Основные принципы работы.....	5
1.5. Архитектурные особенности .....	5
1.6. Ограничения системы .....	5
1.7. Основные объекты системы.....	5
1.8. Целевой результат применения.....	6
2. Авторизация и управление доступом .....	6
2.1 Назначение механизма авторизации .....	6
2.2. Экран авторизации.....	6
2.3. Процесс аутентификации.....	6
2.4. Проверка прав и разграничение доступа.....	7
2.5. Обновление токена .....	7
2.6. Завершение сессии .....	7
2.7. Ролевая модель .....	7
3. Ролевая модель .....	8
3.1. Назначение ролевой модели .....	8
3.2. Классификация ролей .....	8
3.2.1. Системные роли .....	8
3.2.2. Базовая роль.....	8
3.2.3. Функциональные роли .....	8
3.2.4. Контекстная роль .....	9
3.3. Основные принципы разграничения доступа.....	10
3.3.1. Зоны ответственности.....	10
3.3.2. Типы прав.....	10
3.3.3. Наследование ролей.....	10
3.3.4. Режим «только чтение» (read-only) .....	10
3.3.5. Отказ в доступе .....	10
3.4. Применение ролевой модели к объектам системы.....	10
3.4.1. Список заявок.....	10
3.4.2. Карточка заявки .....	11
3.4.3. Комментарии и чаты.....	12
3.4.4. Вложения.....	12
3.4.5. Связанные задачи .....	12
3.4.6. Компании, услуги, продукты .....	12
3.4.7. Журнал активности .....	12
3.4.8. Массовые операции и административные действия .....	12
4. Работа с заявками.....	13
4.1. Общие положения.....	13
4.2. Список заявок .....	13
4.2.1. Общие положения .....	13
4.2.2. Область видимости .....	13
4.2.3. Отображение списка.....	13
4.2.4. Два режима открытия карточки заявки .....	15
4.2.6. Поиск.....	15
4.2.7. Фильтрация.....	16
4.2.8. Сортировка .....	19
4.2.9. Сохранение пользовательских настроек.....	20
4.2.10. Пагинация.....	20

4.2.11. Ошибки и состояния .....	21
Основные состояния: .....	21
Состояние загрузки (skeleton): .....	21
4.3. Карточка заявки .....	21
4.3.1. Структура карточки .....	22
4.3.2. Отображение полей и правила доступности .....	22
4.3.3. Редактирование полей .....	23
4.3.4. Статусы и жизненный цикл заявки .....	23
4.3.5. Вложения .....	23
4.3.6. Связанные заявки .....	23
4.3.7. Комментарии .....	24
4.3.9. История изменений (Activity Log) .....	24
5. Список компаний .....	24
5.1. Функциональные возможности .....	24
5.1.1. Просмотр списка компаний .....	24
5.1.2. Поиск .....	25
5.1.3. Фильтрация .....	25
5.1.4. Сортировка .....	25
5.1.5. Постраничное отображение .....	25
5.1.6. Открытие карточки компании .....	25
5.1.7. Создание компании .....	25
5.2. Ограничения .....	26
6. Карточка компании .....	26
6.1. Основные сведения .....	26
6.2. Услуги компании .....	26
6.3. Контактная информация .....	27
6.4. Дополнительные параметры .....	27
6.5. Работа со статусом компании .....	27
6.6. Связанные объекты .....	27
6.7. Доступные действия .....	27
6.8. Ограничения .....	27
7. Профиль пользователя .....	28
7.1. Структура .....	28
7.2. График работы .....	29
7.3. Управление сессиями .....	29
7.4. Переходы .....	29
7.5. Выход из профиля .....	29

# 1. Общие сведения

## 1.1. Назначение системы

Система Service Desk предназначена для регистрации, обработки и контроля жизненного цикла заявок, связанных с инцидентами, запросами на обслуживание, изменениями, проблемами и другими типами обращений.

Система обеспечивает централизованное ведение обращений, формирование единого хранилища информации, управление связанными объектами и контроль соблюдения установленных сроков обслуживания (SLA).

Система ориентирована на использование в организациях с распределённой структурой, включающих множество подразделений, команд, компаний-клиентов и исполнителей.

Service Desk обеспечивает унифицированный механизм фиксации обращений и предоставляет средства их эффективной маршрутизации, контроля и анализа.

## 1.2. Область применения

Система применяется для:

- обработки пользовательских инцидентов и запросов;
- регистрации обращений сотрудников и внешних клиентов;
- выполнения служебных и эксплуатационных задач;
- предоставления сервисов в рамках действующих соглашений об уровне обслуживания;
- организации взаимодействия между подразделениями и сервисными командами;
- обеспечения аудита, контроля сроков и фиксации всей истории обработки заявок.

Service Desk используется в деятельности технической поддержки, ИТ-подразделений, служб эксплуатации и других команд, участвующих в обработке обращений.

## 1.3. Функциональное назначение

Система включает следующие основные функциональные подсистемы:

- **Авторизация и управление доступом**  
Обеспечивает аутентификацию пользователей, привязку сессий, актуализацию токенов и контроль доступа согласно ролевой модели.
- **Подсистема заявок**  
Предоставляет интерфейсы для регистрации, просмотра, фильтрации, обработки и контроля ошибок по всем обращениями. Поддерживает жизненный цикл, статусы, переходы, комментарии, вложения, связанные задачи и работу в реальном времени.
- **Подсистема компаний**  
Содержит сведения о юридических лицах и организациях, с которыми связаны заявки. Определяет продукты, услуги и конфигурации, влияющие на бизнес-логику обработки обращений.
- **Профиль пользователя**  
Содержит персональные данные, график работы, проекты и активные сессии. Обеспечивает полную управляемость пользовательских параметров в рамках ролевой модели.
- **Справочники и конфигурации**  
Содержат исходные данные, такие как статусы, типы заявок, продукты, услуги, приоритеты, параметры SLA, бизнес-правила, company-config и расширенные атрибуты.
- **Система уведомлений**  
Обеспечивает доставку внутренних уведомлений и информирование об изменениях заявок.

## 1.4. Основные принципы работы

Система использует следующие принципы:

- **Единый жизненный цикл заявок**  
Все обращения проходят унифицированный набор статусов согласно конфигурации компании, включая регистрацию, выполнение, проверку, закрытие и дополнительные этапы.
- **Принцип контекстности данных**  
Доступная пользователю информация зависит от выбранной компании и предоставленных ролей.
- **Разделение прав доступа**  
Поведение интерфейсов, доступность операций и набор данных определяются ролевой моделью.
- **Поддержка SLA**  
Все задачи обрабатываются в соответствии с установленными нормативами времени. Система фиксирует сроки, отклонения и нарушения.
- **Прозрачность выполнения**  
История действий, включая комментарии, изменения статусов, вложения, обновления и переходы между ролями, сохраняется и доступна в карточке заявки.
- **Масштабируемость конфигурации**  
Поведение системы может изменяться в зависимости от параметров `company-config` без необходимости модификации серверного кода.

## 1.5. Архитектурные особенности

Система функционирует по клиент-серверной архитектуре и использует следующие механизмы:

- **Авторизация через Keycloak**  
Применяется OAuth2 + JWT, поддерживаются механизмы обновления токенов, распределённые блокировки при refresh, защита от подмены токенов, использование `httpOnly cookies`.
- **REST API**  
Взаимодействие клиентских интерфейсов с сервером осуществляется через стандартизированные HTTP-запросы.
- **Company-config**  
Каждая компания имеет собственную конфигурацию, определяющую набор полей заявки, статусы, переходы, обязательные поля, SLA, бизнес-правила и ограничения.
- **Хранение сессий**  
Реализовано хранение сеансов пользователей в базе данных с шифрованием refresh-токенов и контролем срока жизни сессии.

## 1.6. Ограничения системы

Система предусматривает следующие ограничения:

- удаление компаний не допускается (используется деактивация);
- удаление заявок не предусмотрено;
- изменение корректных системных конфигураций допускается только через административные роли;
- процессинг приложений производится в пределах доступных типов файлов;
- операции редактирования зависят от ролевой модели и могут быть недоступны для пользователей базовых ролей.

## 1.7. Основные объекты системы

К основным объектам относятся:

- пользователи;
- компании;

- заявки;
- продукты и услуги;
- комментарии и вложения;
- статусы и правила переходов;
- справочники (приоритеты, типы, параметры);
- расписания работы пользователей;
- конфигурации компании.

Каждый объект имеет набор атрибутов и бизнес-правил, определённых конфигурацией.

## 1.8. Целевой результат применения

Использование системы позволяет обеспечить:

- централизованное управление обработкой обращений;
- прозрачность процессов и историю всех действий;
- соблюдение SLA и контроль сроков выполнения;
- корректную маршрутизацию заявок на ответственные команды;
- актуальность данных о пользователях, компаниях и продуктах;
- формирование репозитория данных для аналитической отчетности.

# 2. Авторизация и управление доступом

## 2.1 Назначение механизма авторизации

Механизм авторизации предназначен для идентификации пользователей, проверки их полномочий и обеспечения контролируемого доступа к функциональным компонентам системы.

Авторизация обеспечивает:

- подтверждение личности пользователя на основе предоставленных учётных данных;
- применение ролевой модели и ограничений доступа;
- соблюдение зон ответственности (компаний, подразделений);
- предотвращение несанкционированного доступа к данным;
- выполнение проверок при обращениях к функционалам системы.

Авторизация является обязательной для всех операций, связанных с доступом к данным или функциональным сущностям.

## 2.2. Экран авторизации

Экран авторизации представляет собой интерфейс для ввода учётных данных пользователя. Форма включает:

- поле «логин/электронная почта»;
- поле «пароль»;
- управляющий элемент подтверждения входа.

При некорректных данных система отображает универсальное сообщение об ошибке без раскрытия информации о существовании учетной записи.

Доступ к дальнейшим модулям предоставляется только после успешной авторизации.

## 2.3. Процесс аутентификации

Аутентификация выполняется посредством сервера авторизации Keycloak.

Процесс включает следующие этапы:

1. Пользователь вводит логин и пароль.
2. Клиентская часть направляет запрос на сервер приложения.
3. Сервер приложения передаёт данные в Keycloak для проверки.

4. Keycloak формирует токены (access\_token, refresh\_token) и возвращает их серверу приложения.
5. Сервер приложения:
  - a) валидирует полученные данные;
  - b) создаёт идентификатор пользовательской сессии (session\_id);
  - c) сохраняет сведения о сессии в базе данных, включая зашифрованный refresh\_token.
6. Клиент получает access\_token и session\_id в защищённых httpOnly-cookie.

Токен доступа имеет ограниченный срок действия и используется для выполнения всех запросов к API.

## 2.4. Проверка прав и разграничение доступа

При каждом обращении к системе выполняются следующие проверки:

- проверка подписи access\_token;
- проверка соответствия получатель (aud);
- проверка срока действия токена;
- проверка соответствия session\_id ;
- применение ролевой модели;
- применение ограничений по зоне ответственности пользователя

Доступ пользователя ограничивается:

- ролью,
- зоной ответственности,
- состоянием объекта,
- режимом (read / update / delete / read-in / none)

## 2.5. Обновление токена

Для обеспечения непрерывности работы используется механизм обновления токенов.

Основные положения:

- Access\_token имеет срок действия 5 минут;
- клиент инициирует проверку срока действия на регулярной основе;
- при остаточном времени < 60 секунд выполняется запрос обновления;
- идентификация сессии выполняется по session\_id
- для предотвращения параллельных попыток обновления используется механизм блокировки (distributed lock).

После успешного обновления клиент получает новый access\_token и обновлённый срок действия пользовательской сессии.

## 2.6. Завершение сессии

Сессия пользователя завершается при наступлении одного из условий:

- истечение срока действия refresh\_token;
- достижение максимального времени жизни сессии;
- отсутствие активности пользователя в течение установленного периода;
- явный выход пользователя из системы.

После завершения сессии доступ к функциональности блокируется до повторного прохождения авторизации.

## 2.7. Ролевая модель

Ролевая модель определяет уровень доступа пользователей к объектам и операциям.

Роли разделяются на:

- **системные:**
  - Суперадминистратор
  - Администратор
  - Координатор
- **базовые:**

- Пользователь
- **функциональные** (применяются на уровне конкретной заявки):
  - Ответственный
  - Исполнитель
  - Наблюдатель
- **контекстные**:
  - Инициатор

При наличии нескольких ролей применяется объединённый набор прав с приоритетом расширяющих полномочий.

## 3. Ролевая модель

### 3.1. Назначение ролевой модели

Ролевая модель определяет объём прав пользователей при работе с системой и обеспечивает контролируемый доступ к функциональности, данным и объектам.

Ролевое разграничение используется для:

- ограничения операций в соответствии с полномочиями пользователя;
- определения видимости данных (заявок, компаний, связанных сущностей);
- предотвращения несанкционированного доступа;
- применения функциональных ролей в контексте отдельных объектов (например, заявок);
- соблюдения зон ответственности.

Ролевая модель является обязательным компонентом системы безопасности и применяется при каждом обращении к данным.

### 3.2. Классификация ролей

Все роли в системе распределяются на следующие группы:

#### 3.2.1. Системные роли

Предоставляют расширенные административные полномочия:

- **Суперадминистратор**  
Полный доступ к конфигурации системы, пользователям, зонам, объектам и настройкам.
- **Администратор**  
Управление справочниками, зонами, компаниями, параметрами заявок, ролями пользователей.
- **Координатор**  
Управление процессом обработки заявок в пределах зоны ответственности, контроль сроков и распределение задач.

#### 3.2.2. Базовая роль

- **Пользователь**  
Базовый уровень доступа. Пользователь работает с заявками, в которых участвует, и не имеет прав на изменение системных объектов.

#### 3.2.3. Функциональные роли

Назначаются на уровне конкретной заявки и определяют доступ к действиям внутри неё:

- **Ответственный**  
Лицо, принимающее решение по заявке, определяющее ход её выполнения.
- **Исполнитель**  
Пользователь, выполняющий работы по заявке.

- **Наблюдатель**

Пользователь, подписанный на уведомления и контролирующий ход выполнения, без права менять данные.

Функциональные роли действуют только в контексте конкретной заявки и не расширяют полномочия в системе в целом.

### **3.2.4. Контекстная роль**

- **Инициатор**

Пользователь, создавший заявку. Может просматривать ход её выполнения, комментарии и вложения, но не имеет права редактировать содержание.

## 3.3. Основные принципы разграничения доступа

### 3.3.1. Зоны ответственности

Каждый пользователь привязан к одной или нескольким зонам (компания, подразделение, регион).

Пользователь не видит объекты, находящиеся вне его зоны ответственности.

### 3.3.2. Типы прав

В системе используются следующие типы прав:

- **read** - просмотр данных;
- **update** - создание и изменение данных;
- **delete** - удаление данных;
- **read-in** - просмотр связанной сущности из карточки объекта;
- **none** - отсутствие доступа.

### 3.3.3. Наследование ролей

- При наличии нескольких ролей применяются наиболее расширенные права.
- Функциональные роли (Ответственный, Исполнитель) имеют приоритет над базовой ролью в контексте заявки.
- Удаление функциональной роли приводит к автоматическому снижению доступа до уровня базовой роли.

### 3.3.4. Режим «только чтение» (read-only)

Интерфейс работает в режиме read-only при отсутствии прав update:

- все редактируемые поля блокируются;
- элементы управления скрываются;
- при попытке взаимодействия выводится уведомление «Недостаточно прав».

### 3.3.5. Отказ в доступе

При обращении к объекту вне зоны видимости:

- отображается единый экран «Доступ ограничен»;
- данные не загружаются;
- сервер возвращает код 403.

## 3.4. Применение ролевой модели к объектам системы

Ролевая модель определяет доступ пользователя к объектам системы и перечень разрешённых операций.

Проверка прав выполняется при каждом обращении к объекту, включая операции чтения, изменения, удаления, переходов по ссылкам и взаимодействия со связанными сущностями.

Все ограничения применяются одновременно: по роли, по зоне ответственности и по типу объекта.

### 3.4.1. Список заявок

Для списка заявок ролевой моделью определяется:

- объём отображаемых заявок;
- доступность функций поиска, сортировки и фильтрации;
- возможность настройки параметров отображения;

- возможность перехода в карточку заявки.

Пользователь видит только те заявки, которые соответствуют его зоне ответственности и его роли (инициатор, исполнитель, наблюдатель).

Заявки вне зоны пользователя не отображаются и не участвуют в поиске.

### 3.4.2. Карточка заявки

Доступ к карточке заявки определяется сочетанием ролевых и зональных ограничений:

- роль **Инициатора** позволяет просматривать заявку, но не позволяет изменять её атрибуты;
- функциональные роли **Ответственного** и **Исполнителя** предоставляют права на изменение данных, предусмотренных конфигурацией;
- роль **Наблюдателя** предоставляет доступ только на чтение;
- роль **Пользователя** позволяет просматривать заявки, в которых он участвует;
- системные роли имеют расширенные права согласно их назначению.

Пользователь не может открыть карточку заявки, находящейся вне его зоны ответственности.

В этом случае отображается унифицированный экран «Доступ ограничен» (HTTP 403).

### 3.4.3. Комментарии и чаты

Доступ к комментариям определяется ролевой моделью и контекстной ролью в заявке:

- инициатор имеет доступ к публичному чату с возможностью добавления и редактирования своих сообщений;
- функциональные роли (исполнитель, ответственный) могут добавлять комментарии, прикреплять файлы и взаимодействовать с обеими вкладками чата, при наличии таких прав;
- наблюдатель имеет доступ на чтение к обеим лентам сообщений;
- базовая роль «Пользователь» имеет доступ на чтение, без возможности добавления новых сообщений.

Доступ к внутреннему чату предоставляется только ролям, предусмотренным конфигурацией.

### 3.4.4. Вложения

Поведение при работе с файлами определяется типом роли:

- пользователи с правом update могут загружать, изменять и удалять вложения;
- пользователи с правом read могут просматривать и скачивать вложения;
- пользователи с правом none не видят список вложений;
- пользователи в режиме read-in получают доступ только на просмотр.

Размер вложений ограничен установленным лимитом (50 МБ). Ошибки загрузки и предпросмотра отображаются в виде системных уведомлений.

### 3.4.5. Связанные задачи

Пользователь может просматривать список связанных задач при наличии прав read в карточке заявки:

- переход к связанной сущности возможен только при наличии прав доступа к целевой карточке;
- при отсутствии доступа к связанной сущности отображается экран «Доступ ограничен»;
- создание или удаление связи доступно только пользователям с правами update (в соответствии с ролью).

### 3.4.6. Компании, услуги, продукты

Доступ к связанным объектам (компаниям, услугам, продуктам) осуществляется в режиме read или read-in:

- пользователи базовых и функциональных ролей видят только краткую информацию внутри карточки заявки;
- переход в карточку компании или услуги недоступен при отсутствии прямого разрешения (read-in запрещён);
- системные роли могут выполнять полный набор операций согласно конфигурации.

При попытке прямого перехода к объекту вне зоны ответственности автоматически отображается экран «Доступ ограничен» (HTTP 403).

### 3.4.7. Журнал активности

Доступ к просмотру журнала действий зависит от роли:

- для базовых и функциональных ролей доступ к активности может отсутствовать (none);
- системные роли имеют доступ к полному журналу.

### 3.4.8. Массовые операции и административные действия

Выполнение операций массового изменения (создание, дублирование, мастер-действия, удаление объектов) доступно только системным ролям.

Пользователи базовых и функциональных ролей не имеют доступа к массовым операциям.

## 4. Работа с заявками

### 4.1. Общие положения

Работа с заявками является центральной функциональной областью системы. Все операции по регистрации, анализу, обработке, выполнению и контролю обращений осуществляются в рамках модулей списка заявок и карточки заявки.

Доступ к данным и возможность выполнения действий определяются ролевой моделью, зоной ответственности пользователя и установленными правилами бизнес-логики.

Система обеспечивает:

- отображение заявок в пределах доступной зоны;
- выполнение операций согласно роли пользователя (read, update, delete, read-in, none);
- ведение статуса и жизненного цикла заявки;
- взаимодействие между участниками обработки заявки;
- работу с вложениями, комментариями и связанными сущностями;
- контроль сроков выполнения (SLA) и фиксацию истории изменений.

### 4.2. Список заявок

Раздел «Список заявок» предназначен для централизованного отображения всех заявок, доступных пользователю в рамках его роли и зоны ответственности. Список является точкой входа в карточку заявки, позволяет выполнять поиск, фильтрацию, сортировку, анализ нагрузки и контроль актуального состояния обращений.

#### 4.2.1. Общие положения

Список заявок представляет собой динамическое табличное представление данных, получаемых с сервера по REST API.

Основные принципы работы:

- загрузка списка осуществляется через запрос `GET /tickets`;
- поддерживается пагинация (cursor-based) и загрузка данных по мере прокрутки;
- данные обновляются в режиме реального времени через WebSocket/SSE (при недоступности канала применяется ручное обновление);
- отображаются только те атрибуты, которые разрешены конфигурацией компании (company-config);
- визуальные placeholder-строки (skeleton) используются при загрузке, поиске и перефильтрации.

Заявки, недоступные пользователю по зоне ответственности или ролям, не включаются в список и не участвуют в поиске.

#### 4.2.2. Область видимости

Область видимости определяется:

- ролевой моделью,
- зоной ответственности пользователя,
- ограничениями company-config,
- функциональными ролями по конкретным заявкам.

Пользователь видит только те заявки, по которым у него есть право просмотра.

Попытка получить доступ к недоступной заявке приводит к ошибке 403 и выводу экрана «Доступ ограничен» (даже при прямом переходе по ссылке).

#### 4.2.3. Отображение списка

Список заявок отображается в виде таблицы, содержащей ключевые атрибуты:

- приоритет;
- ключ (человекочитаемый номер, например, INC-25);
- название заявки;
- статус;
- исполнитель;
- опциональные поля: компания, дата создания, срок исполнения, показатели SLA и др.

Список атрибутов и обязательность колонок определяется `company-config`.

Обязательные колонки скрыть невозможно.

Все строки таблицы являются интерактивными.

## 4.2.4. Два режима открытия карточки заявки

Система поддерживает два режима просмотра заявки из списка:

### 1. Встроенный режим (мини-карточка в списке)

Открывается при выборе строки по любому полю, **кроме ключа заявки**.

Характеристики:

- карточка отображается в виде боковой панели (Sneak Peek) справа от списка;
- показывает основные свойства заявки: заголовок, ключ, статус, исполнитель, приоритет, сроки и описание;
- не прерывает работу со списком: фильтры, сортировки и позиция прокрутки сохраняются;
- позволяет переходить к другой мини-карточке простым выбором новой строки;
- доступность данных регулируется ролевой моделью и зоной ответственности.

Если прав на просмотр нет — отображается stub «Недостаточно прав для просмотра».

### 2. Полноэкранная карточка заявки

Открывается при клике **по ключу заявки** (например, «INC-25»), либо по кнопке перехода внутри мини-карточки.

Характеристики:

- выполняется переход на страницу /tickets/{number}
- отображается полный состав данных, включая историю активности, вложения, связанные сущности и внутренний чат;
- доступны все разрешённые действия (изменение статуса, добавление комментариев, вложений и др.);
- все проверки доступа выполняются на уровне API.

Полноэкранный режим используется для работы с заявкой, требующей детального анализа.

## 4.2.6. Поиск

Поиск является основным инструментом текстового поиска по доступным пользователю заявкам. Механизм поиска работает совместно с активными фильтрами и ограничениями ролевой модели.

#### 4.2.6.1. Область поиска

Поиск выполняется:

- по ключу заявки (например: INC-25);
- по названию заявки (частичное вхождение, поиск по подстроке);
- по текстовым полям, разрешённым конфигурацией компании.

Поиск применяется **только к заявкам, находящимся в области видимости пользователя.**

#### 4.2.6.2. Принцип работы

При вводе текста:

1. система выполняет поиск с задержкой (debounce);
2. список переходит в состояние загрузки (skeleton);
3. выполняется запрос к API с учётом поискового слова и текущих фильтров;
4. отображается результат поиска.

Поиск всегда выполняется совместно с фильтрами (логика AND).

#### 4.2.6.3. Поведение при отсутствии результатов

Если по результатам поиска нет совпадений:

- отображается экран состояния «Поиск не дал результатов»;
- выводится системное сообщение:  
**«Поиск не дал результатов. Попробуйте снова, изменив параметры запроса или сбросьте фильтры.»**
- отображается кнопка:  
**«Сбросить фильтры»**

Нажатие кнопки:

- очищает строку поиска;
- снимает все активные фильтры;
- загружает список в состоянии по умолчанию.

Это состояние применяется **только при поисковом запросе**, а не при обычной фильтрации.

#### 4.2.6.4. Интеграция с фильтрами

- Поиск работает поверх всех выбранных фильтров.
- Изменение фильтров автоматически сбрасывает состояние пустого поиска и инициирует новое обновление списка.
- Если активные фильтры ограничивают список до пустого состояния **без поискового запроса**, показывается другое состояние («Заявки не найдены») — поведение отличается от пустого результата поиска.

#### 4.2.6.5. Интеграция с мини-карточкой заявки

- Открытие мини-карточки не сбрасывает поисковый запрос.
- Закрытие мини-карточки также не изменяет состояние поиска.
- При выборе строки внутри результатов поиска отображаются только найденные заявки.

#### 4.2.6.6. Ограничения доступа

Если пользователь вводит ключ существующей заявки, к которой у него нет прав:

- заявка не отображается в результатах поиска;
- состояние отображается как «Поиск не дал результатов»;
- система не раскрывает факт существования скрытой заявки.

#### 4.2.7. Фильтрация

Фильтрация является ключевым механизмом формирования списка заявок. Все фильтры отображаются в виде панели-контейнера над таблицей и применяются в режиме реального времени.

Система поддерживает несколько типов фильтров, включая множественный выбор значений.

#### **4.2.7.1. Панель фильтров**

Панель фильтров включает:

- фиксированные фильтры (например: Тип заявки, Исполнитель, Приоритет, Статус);
- пользовательские фильтры, добавляемые через кнопку «+»;
- кнопку сброса всех фильтров;
- выпадающий список «Мои фильтры».

Панель автоматически переносит элементы на вторую строку при большом количестве активных фильтров.

#### 4.2.7.2. Механизм фильтрации

##### Основные правила

- Фильтрация выполняется на сервере.
- Значения фильтров ограничиваются областью видимости пользователя.
- Комбинация фильтров работает в режиме AND (пересечение условий).
- Фильтры не раскрывают недоступные знания, т. е. пользователь не видит значения справочников, не относящиеся к его зоне ответственности.

##### Интерактивность

- Каждое изменение фильтра приводит к перезагрузке списка:
  - отображается состояние skeleton;
  - после загрузки показывается обновлённый набор строк.

#### 4.2.7.3. Фильтры с множественным выбором

- **Тип заявки**  
(Инцидент, Проблема, Запрос на обслуживание, Запрос на изменение)
- **Исполнитель**  
(список пользователей текущей зоны)
- **Приоритет**  
(Низкий, Средний, Высокий, Наивысший)
- **Статус**  
(Новый, Назначено, В работе, Зарегистрировано, Закрыто, Ожидает ответа инициатора и др.)

Особенности:

- Допускается выбор нескольких элементов одного фильтра.
- Активные значения отображаются внутри блока фильтра в виде бейджей.
- При выборе большого количества значений блок фильтра расширяется.

#### 4.2.7.4. Пользовательские фильтры

Пользователь может:

- сохранить текущую комбинацию фильтров под собственным именем;
- выбрать ранее сохранённый фильтр из списка «Мои фильтры»;
- удалить сохранённый фильтр.

Создание пользовательского фильтра выполняется через модальное окно:

**Название фильтра** → **Создать**

Доступ к фильтрам сохраняется между сессиями пользователя, согласно механизму хранения пользовательских настроек.

#### 4.2.7.5. Динамическое добавление фильтров

Кнопка «+» открывает меню доступных фильтров, позволяющее:

- добавить дополнительные критерии;
- включать / выключать параметры фильтрации (например: Тип заявки, Исполнитель, Статус, Дата создания).

После добавления фильтр отображается в панели и может быть удалён пользователем.

#### 4.2.7.6. Отображение результатов

После применения фильтров:

- отображается только набор заявок, соответствующих всем активным критериям;
- неактуальные строки скрываются полностью;
- на панели таблицы отображается количество найденных записей.

#### 4.2.7.7. Сброс фильтров

Панель «Мои фильтры» содержит встроенную кнопку для полного сброса:

##### **Сбросить фильтры**

После сброса:

- панель фильтров возвращается в состояние по умолчанию;
- список заявок перезагружается;
- интерфейс очищает бейджи активных значений.

#### 4.2.7.8. Пагинация

Пагинация расположена под списком заявок:

- переключение осуществляется по страницам (1–2–3–4...);
- размер страницы выбирается пользователем (25 / 50 / 100 записей);
- изменение размера страницы приводит к перезагрузке списка.

Пагинация не зависит от выбранных фильтров и работает поверх уже отфильтрованных данных.

#### 4.2.8. Сортировка

Сортировка предназначена для упорядочивания отображаемых заявок по одному из доступных атрибутов. Механизм сортировки позволяет пользователю быстро изменять порядок отображения данных, не меняя текущих фильтров и состояния поиска.

##### 4.2.8.1. Доступные поля сортировки

Сортировка может быть применена к любому полю таблицы, которое в конфигурации компании обозначено как `sortable`.

Типовые `sortable`-поля:

- ключ заявки (идентификатор);
- название заявки;
- приоритет;
- статус;
- исполнитель;
- дата создания.

Список поддерживаемых атрибутов определяется конфигурацией `company-config` и может отличаться между организациями.

##### 4.2.8.2. Принцип работы сортировки

- Сортировка выполняется на серверной стороне и применяется ко всему набору данных, соответствующему активным фильтрам и поисковому запросу.
- Поддерживается однопараметрическая сортировка (только по одному полю одновременно).
- Повторное нажатие на заголовок колонки изменяет направление сортировки:

- по возрастанию → по убыванию → сброс сортировки.
- После изменения сортировки таблица переводится в состояние загрузки (skeleton) и обновляет результат.

#### 4.2.8.3. Особенности сортировки для полей с перечислимым типом

Для полей, у которых значения являются перечислениями (enum), сортировка выполняется в соответствии с порядком, заданным в конфигурации системы.

Типичные примеры:

##### Приоритет

Порядок сортировки определяется заранее установленным ранжированием:

Низкий → Средний → Высокий → Наивысший.

##### Статус

Порядок определяется схемой жизненного цикла заявки. Например:

Новый → Назначено → В работе → Ожидает ответа инициатора → Закрыто.

Эти порядки могут быть изменены администратором через настройки компании.

#### 4.2.8.4. Совместимость с фильтрацией и поиском

- Сортировка применяется **поверх текущих фильтров**.
- При изменении фильтра сортировка сохраняется.
- При выполнении поиска сортировка также остаётся активной и применяется к найденным данным.
- При сбросе фильтров сортировка **не сбрасывается** автоматически (сохраняется до прямого изменения пользователем).

#### 4.2.8.5. Сохранение состояния сортировки

Механизм персональных настроек пользователя сохраняет выбранное состояние сортировки:

- выбранную колонку;
- направление сортировки.

При последующих открытиях списка заявок сортировка применяется автоматически (если не была изменена вручную).

Логика сохранения описана в разделе 4.2.9 «Сохранение пользовательских настроек».

#### 4.2.9. Сохранение пользовательских настроек

Система поддерживает автоматическое сохранение пользовательских настроек отображения списка заявок. Настройки привязаны к учетной записи пользователя и восстанавливаются при последующем входе в систему.

Сохраняются следующие параметры:

- выбранные фильтры и их значения, включая множественный выбор;
- выбранный фильтр из раздела «Мои фильтры»;
- набор отображаемых колонок и их порядок;
- состояние сортировки;
- выбранный размер страницы (25 / 50 / 100 записей).

Сохранение выполняется автоматически при изменении любого параметра списка.

При повторном открытии раздела система применяет ранее сохранённое состояние с учетом ограничений компании и ролевой модели (недоступные значения фильтров автоматически исключаются).

Строка поиска и состояние открытой мини-карточки не сохраняются между сессиями.

#### 4.2.10. Пагинация

Система поддерживает постраничный режим отображения списка заявок. Пагинация применяется после всех активных условий поиска и фильтрации.

Основные положения:

- пользователь может выбрать количество записей на странице: **25 / 50 / 100**;
- переключение страниц осуществляется через навигационные элементы в нижней части списка («1 — 2 — 3 — ...»);
- изменение размера страницы приводит к перезагрузке данных с учётом выбранных фильтров и сортировки;
- состояние выбранного размера страницы сохраняется в пользовательских настройках;
- при смене фильтров или поискового запроса пагинация сбрасывается на первую страницу.

Пагинация применяется только к доступному пользователю набору данных и не раскрывает информацию о количестве скрытых заявок вне зоны его ответственности.

#### 4.2.11. Ошибки и состояния

Список заявок поддерживает отображение стандартных системных состояний, возникающих при работе поиска, фильтров и загрузки данных. Каждое состояние отображается в виде отдельного экрана с текстовым сообщением и, при необходимости, управляющими элементами.

##### Основные состояния:

- **«Не создано ни одной заявки»**  
Отображается, если в системе отсутствуют заявки, доступные пользователю.
- **«Заявки не найдены»**  
Отображается при отсутствии результатов по активным фильтрам (при пустой строке поиска).
- **«Поиск не дал результатов»**  
Отображается только при наличии поискового запроса.  
Сопровождается рекомендацией изменить параметры поиска и кнопкой «Сбросить фильтры».
- **«Ошибка при загрузке списка заявок»**  
Используется при недоступности сервера или сетевой ошибке.  
Пользователю предлагается повторить загрузку.
- **«Доступ ограничен»**  
Отображается при попытке перейти к заявке вне зоны ответственности пользователя или при недостаточности прав.

##### Состояние загрузки (skeleton):

Используется:

- при первичной загрузке списка;
- при применении поиска;
- при изменении фильтров;
- при изменении сортировки;
- при переключении страниц.

Skeleton заменяет строки таблицы до получения данных с сервера и предотвращает «скачки» верстки.

### 4.3. Карточка заявки

Карточка заявки является основной формой представления информации о конкретной заявке и используется для просмотра, редактирования, согласования и выполнения работ в рамках установленного жизненного цикла. Карточка обеспечивает доступ ко всем атрибутам заявки, связанным объектам, комментариям, файлам и истории изменений.

Структура и поведение карточки полностью определяются **конфигурацией компании (company-config)** и ролевой моделью доступа.

Отображаемые поля, их обязательность, доступность для редактирования и набор возможных статусов идентифицируются конфигурационными правилами.

### 4.3.1. Структура карточки

Карточка заявки состоит из следующих функциональных областей:

#### 1. Заголовок

Содержит отображаемое имя заявки. Формируется на основании полей title и key. Изменение названия доступно только при наличии соответствующих прав.

#### 2. Блок основных атрибутов

Представляет ключевые параметры заявки в табличном представлении.

Сюда входят (в зависимости от конфигурации):

- Ключ (key),
- статус,
- тип заявки,
- приоритет,
- исполнитель,
- создатель,
- наблюдатели,
- критичность, влияние, массовость, SLA, регион, продукт, услуга, компания,
- даты создания и обновления,
- крайний срок и другие динамические поля (extended attributes).

#### 3. Описание заявки

Текстовое поле с произвольным содержимым.

Может быть обязательным или необязательным в зависимости от конфигурации (required: true/false).

#### 4. вложения

Поддержка загрузки файлов с последующим отображением списка вложений. Возможны ограничения по размерам и типам файлов согласно конфигурации компании.

#### 5. Связанные заявки

Механизм связывания заявок по схеме «родитель — потомок» или произвольная связь (в зависимости от company-config).

Система обеспечивает проверку циклических связей и корректности статусов.

#### 6. Комментарии

Поддерживается несколько типов комментариев:

- общий чат,
- внутренний (служебный) чат, если он включён конфигурацией.

Комментарии могут содержать упоминания, угрозы, вложения и системные уведомления.

#### 7. История изменений (Активность)

Полный журнал действий по заявке: редактирование полей, изменения статусов, добавление связей, комментарии, вложения.

История является неизменяемой.

### 4.3.2. Отображение полей и правила доступности

Каждое поле карточки имеет ряд параметров, определяемых конфигурацией:

- **visible** — отображается ли поле для данной компании;
- **editable** — доступно ли поле для редактирования (с учётом роли пользователя);
- **required** — является ли поле обязательным для сохранения;
- **type** — тип данных (enum, string, user, company, datetime, multiselect и др.);
- **options** — справочник значений (для enum);
- **depends\_on** — наличие зависимостей от других полей;
- **validation** — набор ограничений;
- **config\_version** — версия конфигурации компании, по которой отрисована карточка.

Доступ пользователя к редактированию полей определяется:

- базовой ролевой моделью,

- схемой жизненного цикла статусов,
- типом поля,
- компанией, к которой относится заявка.

### 4.3.3. Редактирование полей

Редактирование полей выполняется в inline-режиме. При изменении поля:

- выполняется локальная валидация (клиентская);
- отправляется запрос на сервер;
- сервер выполняет бизнес-валидацию согласно company-config;
- поле обновляется или возвращается ошибка.

Взаимодействие ограничивается:

- обязательностью поля;
- связями depends\_on;
- правилами перехода (editable\_in\_status)
- типом поля (например, одиночные/множественные значения);
- доступом пользователя.

При ошибке валидации сервер возвращает код 422 с описанием нарушенного правила.

### 4.3.4. Статусы и жизненный цикл заявки

Изменение статуса контролируется схемой переходов (workflow), определённой в конфигурации компании.

Переход возможен, если выполняются все условия:

- статус присутствует в списке допустимых переходов allowed\_transitions
- пользователь обладает необходимой ролью;
- заполнены все обязательные поля;
- соблюдены дополнительные ограничения (например, указано завершение работ).

Ошибка при нарушении правил перехода возвращается также кодом 422.

### 4.3.5. Вложения

Поддерживаются операции:

- загрузка новых файлов;
- удаление файлов (при наличии прав);
- предпросмотр;
- скачивание.

Параметры ограничений:

- максимальный размер файла;
- допустимые MIME-типы;
- максимально допустимое число вложений.

Все ограничения берутся из company-config.

### 4.3.6. Связанные заявки

Связанные заявки отображаются в виде списка с указанием типа связи.

Система контролирует:

- невозможность создания циклических связей;
- достаточность прав пользователя;
- корректность статусов между связанными заявками.

### 4.3.7. Комментарии

Комментарии делятся на:

- **внешние** (видят все заинтересованные стороны),
- **внутренние** (ограничены сотрудниками или ролью).

Поддерживаются:

- текст,
- файлы,
- упоминания пользователей,
- системные уведомления.

Редактирование и удаление комментариев определяется ролевой моделью.

### 4.3.9. История изменений (Activity Log)

История фиксирует все изменения, включая:

- смену статусов,
- изменение полей,
- создание/удаление связей,
- отправку комментариев,
- действия нескольких пользователей,
- системные события.

История формируется на сервере, является неизменяемой и отображается в хронологическом порядке.

## 5. Список компаний

Раздел «Компании» предназначен для отображения и ведения сведений о юридических лицах и организациях, участвующих в процессах обработки заявок.

Раздел обеспечивает централизованное хранение данных о компаниях, используемых при регистрации и маршрутизации обращений, а также при формировании аналитической информации.

### 5.1. Функциональные возможности

#### 5.1.1. Просмотр списка компаний

Пользователь имеет возможность просматривать перечень всех компаний, зарегистрированных в системе. Табличное представление поддерживает горизонтальный скролл при большом числе колонок.

### **5.1.2. Поиск**

В верхней части раздела предусмотрено поле поиска.

Поиск осуществляется по частичному совпадению текста и может учитывать следующие атрибуты:

- наименование компании;
- ИНН;
- КПП;
- другие текстовые поля, отображаемые в таблице.

### **5.1.3. Фильтрация**

Для уточнения выборки предусмотрены фильтры.

Доступные параметры фильтрации:

- тип компании;
- статус компании;
- дополнительные параметры, определённые справочниками.

Фильтры могут комбинироваться между собой.

### **5.1.4. Сортировка**

Сортировка может выполняться по основным колонкам таблицы, в том числе:

- наименование;
- ИНН;
- дата создания;
- статус.

Поддерживаются режимы сортировки по возрастанию и по убыванию.

### **5.1.5. Постраничное отображение**

При превышении установленного лимита отображения используется пагинация.

Поддерживается выбор количества строк на странице.

### **5.1.6. Открытие карточки компании**

Выбор строки в таблице открывает карточку соответствующей компании в полноэкранном режиме.

### **5.1.7. Создание компании**

Пользователям, обладающим соответствующими правами, доступна функция создания новой компании через кнопку «Создать компанию».

## 5.2. Ограничения

- Удаление компаний не предусмотрено; для исключения из использования применяется перевод в статус «Неактивна».
- Изменение информации о компании доступно только ролям, имеющим административные полномочия.
- Неактивные компании могут быть скрыты в фильтрах и недоступны для выбора в связанных разделах, если это предусмотрено настройками.

## 6. Карточка компании

Карточка компании предназначена для просмотра и редактирования сведений о конкретной организации, зарегистрированной в системе.

Карточка обеспечивает централизованный доступ к атрибутам компании, связанной справочной информации и действиям, которые допускаются ролевой моделью.

### 6.1. Основные сведения

В разделе основных сведений отображаются ключевые идентификационные данные компании:

- наименование компании;
- ИНН;
- КПП;
- тип компании;
- статус активности (Активна / Неактивна);
- дата создания записи;
- дата последнего обновления;
- пользователь, создавший запись;
- пользователь, обновивший запись.

Редактирование данных доступно в соответствии с ролевой моделью.

### 6.2. Услуги компании

Карточка содержит перечень услуг, которые предоставляются данной компанией или ассоциированы с ней.

Для блока услуг поддерживаются следующие операции:

- просмотр списка услуг;
- добавление услуги;
- удаление услуги из списка;
- редактирование параметров услуги (при наличии функционала).

## 6.3. Контактная информация

В карточке отображается набор контактных данных компании, включая:

- юридический адрес;
- фактический адрес;
- телефоны;
- электронную почту;
- контактное лицо (если предусмотрено моделью данных).

Каждый параметр может быть изменён пользователями с соответствующими правами.

## 6.4. Дополнительные параметры

Карточка поддерживает расширяемый набор дополнительных полей, определённых бизнес-логикой.

В перечень могут входить:

- регион обслуживания;
- дочерние структуры;
- внутренние классификаторы;
- параметры интеграции (если используются).

Дополнительные параметры доступны для редактирования в соответствии с ролевой моделью.

## 6.5. Работа со статусом компании

Поле статуса позволяет:

- активировать компанию;
- деактивировать компанию.

Деактивация используется вместо удаления и приводит к недоступности компании при создании новых заявок и в связанных справочниках.

## 6.6. Связанные объекты

При наличии функциональности карточка может содержать данные о связанных сущностях:

- заявки, прикреплённые к компании;
- подразделения;
- контактные лица;
- услуги.

Объекты отображаются в виде списка, переход по строке открывает соответствующую карточку объекта.

## 6.7. Доступные действия

В зависимости от роли пользователя доступны следующие операции:

- редактирование сведений о компании;
- изменение статуса;
- управление услугами;
- редактирование контактной информации;
- добавление / удаление дополнительных параметров.

Для пользователей без прав редактирования карточка открывается в режиме «только просмотр».

## 6.8. Ограничения

- удаление компаний не предусмотрено;
- изменения фиксируются в служебных полях «Дата обновления» и «Обновил»;
- при изменении ИНН или КПП могут применяться встроенные валидационные проверки.

## 7. Профиль пользователя

Раздел предназначен для отображения персональных данных пользователя, его графика работы, связанных проектов и активных сессий. Доступ определяется ролевой моделью.

### 7.1. Структура

Профиль включает следующие блоки:

- **Карточка пользователя**

Отображаются: ФИО, должность, подразделение, контактная информация, аватар.

- **Общая информация**

Содержит:

- руководитель;
- график работы (постоянный или гибкий).

- **Проекты**

Список проектов, в которых участвует пользователь.

- **Активные сессии**

Перечень текущих сессий пользователя с указанием устройства, времени активности и отметкой текущей сессии.

## **7.2. График работы**

График работы редактируется через интерфейс выбора расписания.

Поддерживаются:

- постоянный график (единый интервал времени),
- гибкий график (интервалы по дням недели).

Проверки:

- корректность формата времени;
- время начала < времени окончания;
- наличие выбранных дней.

Сохранение выполняется через обновление данных пользователя.

## **7.3. Управление сессиями**

Раздел позволяет завершить все активные сессии пользователя, кроме текущей.

После завершения выполняется обновление списка.

При ошибке отображается уведомление.

## **7.4. Переходы**

Переход к профилю руководителя выполняется при наличии прав.

Профиль руководителя открывается в режиме просмотра без возможности управления сессиями.

## **7.5. Выход из профиля**

При выходе осуществляется инвалидирование сессии и удаление пользовательских токенов.

Пользователь перенаправляется на экран авторизации.