**Техническое задание на выполнение работ по созданию автоматизированной информационной системы “Бизнес-онлайн”**

**Содержание**

[**Определения, обозначения и сокращения**](#_oyi4gwxaf0it) **3**

[**1. Назначение и цели создания системы**](#_dk2hi0r86mzh) **3**

[1.1. Назначение системы](#_wxtatzdacj71) 3

[1.2. Цели создания системы](#_anev4znkgsxi) 4

[1.3. Целевая аудитория](#_j0pnj7j8ojl7) 4

[1.4. Решаемые задачи](#_b3oc4curyjh4) 4

[**2. Структура и функционирование системы**](#_p6s144ehd5tu) **4**

[2.1. Модуль «Авторизация»](#_2uxe5ylmd09a) 5

[2.2. Модуль «Работа с документами»](#_j21bqz7gxcto) 6

[2.2.1. Просмотр перечня доступных к заполнению документов](#_7cudzs5kjc99) 6

[2.2.2. Работа с формами документов](#_gcmr7m57nduf) 6

[2.2.3 Выбор способа подачи документов](#_wkkzbcxtc6hr) 6

[2.3. Модуль «Панель администратора»](#_dbibbndgsfi) 7

[2.3.1. Управление учетными записями пользователей в Системе](#_zf9i1uic65hm) 7

[2.3.2. Настройка прав доступа ролей](#_6xkcbxhhedog) 8

[2.3.3. Работа с конструктором форм отчетности](#_andzrtwrq518) 8

[2.3.4. Просмотр истории действий пользователей](#_cm7b3wfufw2c) 8

[2.4. Модуль «Управление юристами»](#_gobfv944n7mm) 8

[2.4.1. Управление учетными записями юристов в Системе](#_zby2z7cjqmif) 8

[2.4.2. Просмотр истории действий в кабинете юриста](#_hg6cuqec46sp) 8

[2.5. Модуль «Рабочее место юриста»](#_1vmsqrq4otww) 8

[2.5.1. Внесение данных о клиенте в систему (в карточку клиента)](#_eh5umsw99fnf) 9

[2.5.2. Создание задачи](#_9vurtbejzkfe) 10

[2.5.3. Работа с карточками клиентов](#_tfk5yk6atrpc) 11

[2.5.4. Работа с реестром задач](#_fvex87nsvzd1) 12

[2.5.5. Работа с задачами](#_k6bh7j2l931r) 13

[2.5.6. Просмотр истории ранее совершенных действий с документами](#_lugtodmqevwh) 13

[2.5.7. Подписание документов ЭЦП](#_jbwmeps9guq7) 13

[2.5.8. Направление документов в ФНС](#_dutmo03it7g0) 13

[2.5.9. Направление запроса на выпуск ЭЦП в Удостоверяющий центр](#_eq6x7p1spwm7) 13

[2.5.10. Скачивание и распечатка документов](#_vqrn35spcsuj) 14

[2.6. Модуль “Конструктор форм документов”](#_chtihwo9cp2o) 14

[2.6.1. Создание формы документа](#_l2a3cw5z0zh) 14

[2.6.2. Привязка полей формы к PDF документу](#_pkhcp6o50kf8) 16

[2.6.3. Изменение формы документа](#_h29qq1ueslqt) 16

[2.7. Модуль “Логирование и уведомления”](#_r4adywods0wq) 16

[2.7.1. Автоматическое логирование действий](#_k8jjt1pqf3kt) 16

[2.7.2. Просмотр истории залогированных действий](#_v8xcixsgyysb) 17

[2.7.3. Просмотр уведомлений](#_9uy6hewtggib) 17

[2.8. Модуль “Главная страница”](#_7fhdnazctxmt) 17

[2.8.1. Набор информации для роли “Администратор”](#_m5wv0tahs949) 17

[2.8.2. Набор информации для роли “Руководитель компании”](#_84ayz9hki0xx) 17

[2.8.3. Набор информации для роли “Юрист”](#_8x2757uky3h6) 17

[**3. Ролевая структура системы**](#_mjbkvka2wvyj) **18**

[**4. Организация хранения данных**](#_148so4jblajx) **18**

[**5. Интеграции**](#_eef9w0tltlpi) **18**

[5.1. Интеграция с ФНС](#_93gv0slys08r) 19

[5.2. Интеграция с Dadata](#_sw73ls67mkfy) 19

[5.2.1 Валидация адреса](#_1dqhvdehvfn5) 19

[5.2.2 Подсказки по адресам](#_kjx9dvvfs37b) 19

[5.2.3 Стандартизация адреса](#_9x4qarz6nn3) 22

[5.2.4 Поиск компании по ИНН/ОГРН](#_ahc29rdnhj32) 24

[5.3. Интеграция с УЦ “Основание”](#_4lpgb6fki2tt) 29

[5.3.1 Создание заявки](#_7b3i92qly08n) 29

[5.3.2 Изменение данных по заявке](#_n59yhwyi7k3k) 32

[5.3.3 Перемещение заяви в архив](#_1ejc71xb8m43) 32

[5.3.4 Просмотр сведений о существующей заявке](#_b4edxnba07dk) 32

[5.3.5 Прикрепление документов к существующей заявке](#_jrwnbfz2emkh) 33

[5.3.6 Получение списка продуктов](#_mmyckhmy1to1) 34

[5.4. Подпись контейнера документов через КриптоПро](#_hxf273ipad71) 34

[5.4.1 Подготовка к подписанию документа](#_gku0wwud2l65) 34

[5.5. Интеграция с myDSS](#_grvky0qqx5uo) 35

[**6. Схема бизнес-процессов**](#_8pm4ndeo50ev) **35**

# **Определения, обозначения и сокращения**

Перечень определений, обозначений и сокращений, используемых в настоящем техническом задании, указан в таблице 1.

Таблица 1. Определения, обозначения и сокращения

| **Сокращение (обозначение)** | **Значение сокращения (обозначения)** |
| --- | --- |
| Документ | Результат заполнения шаблона документа в системе |
| Задача | Создаваемая в Системе задача на оказание услуги (регистрации юридического лица или внесение изменений в его регистрационные данные; заявка на выпуск ЭП) |
| Поле | Компонент, представляющий структуру таблицы |
| Пользователь | Пользователь разрабатываемой системы, обладающий правами в соответствии с ролевой моделью |
| Система | Разрабатываемая автоматизированная информационная система “Бизнес-онлайн” |
| УЦ | Удостоверяющий центр |
| ФНС | Федеральная Налоговая Служба |
| Шаблон формы документа | Веб-версия документа, которую заполняет пользователь и которая впоследствии привязывается к конечному документу для отправки или печати |
| Элемент конструктора форм отчетности | Составная часть формы документа, доступная к добавлению через конструктор: заголовок, параметр, разделитель и прочее |
| ЭП | Электронная подпись |

# **1. Назначение и цели создания системы**

## **1.1. Назначение системы**

Система предназначена для автоматизации работы с документами и создания заявки в УЦ “Основание” на создание сертификата ЭП и должна обеспечивать:

* инструмент для создания документов для регистрации юридического лица или внесения изменений в его регистрационные данные;
* оперативный обмен данными между клиентами, юристами, ФНС и УЦ “Основание”;
* бесшовную интеграцию информационных систем внутри проекта, путем использования технологии передачи данных API;
* систематизацию хранения документов;
* доступ пользователей к документам с помощью веб-интерфейса.

## **1.2. Цели создания системы**

Целями создания системы являются:

* создание единого информационного пространства, обеспечивающего весь цикл работы с документами для регистрационных действий с юридическими лицами;
* обеспечение взаимодействия с УЦ “Основание” для создания заявок на выпуск ЭП;
* обеспечение надежного хранения документов с соблюдением процедур разграничения доступа.

## **1.3. Целевая аудитория**

Целевой аудиторией системы являются:

* юридические организации – заключают договор с правообладателем системы на подключение к ней юристов организации;
* работающие в организациях юристы – непосредственно работают с системой;
* клиенты – предприниматели, заинтересованные в регистрации юридического лица, внесении изменений о нем, а также в получении ЭП.

## **1.4. Решаемые задачи**

В задачи реализации проекта входят:

* разработка архитектуры, интерфейса и структуры Системы;
* проектирование и внедрение ролевой модели доступа к данным;
* организация хранения данных;
* обеспечение необходимых интеграций.

# **2. Структура и функционирование системы**

Система представляет собой веб-интерфейс, расположенный по выбранному заказчиком адресу.

Программная разработка интерфейса Системы осуществляется на основании макетов экранных форм, утвержденных Заказчиком. Для системы должна быть выполнена верстка под наиболее часто используемые разрешения компьютеров: 1920х1080 и 1366х768 точек. Интерфейс конечного пользователя Системы должен быть русскоязычным.

Для получения доступа к системе пользователям необходимо пройти процедуру авторизации.

Система должна включать в себя следующие функциональные модули:

* модуль «Авторизация»;
* модуль «Работа с документами»;
* модуль «Панель администратора»;
* модуль «Управление юристами»;
* модуль «Рабочее место юриста»;
* модуль “Конструктор форм отчетности”;
* модуль “Логирование и уведомления”;
* модуль “Рабочий стол”.

## **2.1. Модуль «Авторизация»**

Модуль предназначен для авторизации пользователей в Системе и должен предоставлять следующие функциональные возможности:

– Возможность авторизации в системе. Доступ к системе должен осуществляться путем двухфакторной аутентификации:

* Корректного заполнения в форме авторизации пары параметров: логин/пароль; после заполнения формы авторизации должна происходить проверка соответствия введенного пользователем пароля учетной записи в базе данных. В качестве логина может быть использована электронная почта пользователя или его телефон.
* Ввода одноразового СМС кода по номеру телефона. После проверки корректности ввода пары параметров логин/пароль, пользователю на телефон, указанный при создании учетной записи, направляется одноразовый код, который необходимо ввести в соответствующее поле на странице авторизации. После ввода кода должна происходить проверка его соответствия коду, направленному по номеру телефона пользователя.

При совпадении введенных данных с эталонными на обоих этапах аутентификация признается успешной.

При неудачной попытке авторизации на любом из двух этапов появляется сообщение для повторной попытки авторизоваться с формой авторизации.

Под формой авторизации должна быть размещена кнопка восстановления пароля. Форма «Восстановление пароля» должна содержать поле “Электронная почта пользователя, указанная при регистрации”. При использовании данной функции на указанную почту пользователю должна быть отправлена ссылка для восстановления доступа.

## **2.2. Модуль «Работа с документами»**

Модуль предназначен для работы (создание, изменение, удаление) пользователей с документами и должен предоставлять следующие функциональные возможности:

### **2.2.1. Просмотр перечня доступных к заполнению документов**

Для пользователя должен быть доступен перечень документов к заполнению (на регистрацию юридического лица, на внесение изменений и прочих).

От списка документов должна быть реализована возможность перехода к формам заполнения документов. Формы документов должны быть созданы в системе с использованием модуля “Конструктор форм отчетности”, описанного в пункте 2.6 настоящего документа.

### **2.2.2. Работа с формами документов**

Под работой понимается заполнение, изменение и удаление заполненных с помощью форм отчетности документов.

Под заполнением понимается пошаговый ввод пользователем данных клиента. Поля форм заполнения должны быть реализованы в интерфейсе системы в формате пошаговой анкеты для заполнения.

Формат ввода каждого поля предусмотрен формой.

В модуле должно быть реализовано автоматическое заполнение регистрационных данных ЕГРЮЛ / ЕГРИП при вводе ИНН клиента, а также автоматическое заполнение сведений об адресе. Для этого должна быть организована интеграция с внешним источником данных – Dadata. Интеграция описана в пункте 5.2 настоящего документа.

Для документов должна быть предусмотрена возможность сохранения промежуточного результата заполнения (черновика).

### **2.2.3 Выбор способа подачи документов**

После заполнения полей и их проверки должен генерироваться документ, для которого должны быть доступны следующие действия:

б) подписать действующей ЭП на токене и отправить контейнер в ФНС;

в) сохранить и распечатать документы.

## **2.3. Модуль «Панель администратора»**

Модуль предназначен для работы с учетными записями и конструктором шаблонов и должен предоставлять следующие функциональные возможности:

### **2.3.1. Управление учетными записями пользователей в Системе**

Под управлением понимается работа с учетными записями пользователей (создание, редактирование, удаление), а также назначение и разграничение их прав, позволяющих выполнять те или иные функции в Системе.

При этом в Системе должна быть реализована ролевая структура, приведенная в пункте 3 настоящего документа.

В модуле должен содержаться перечень учетных записей пользователей Системы, разделенный на две группы: компании и юристы. При работе с указанным перечнем для каждого пользователя должны быть определены:

– название компании (для компаний);

– ФИО ответственного (для компаний);

– компания, к которой привязан пользователь (для юристов);

– ФИО пользователя (для юристов);

– роль пользователя;

– электронная почта;

– ID компании / юриста;

– дата регистрации;

– дата окончания доступа.

В рамках обеспечения взаимодействия пользователя с перечнем учетных записей модуль предоставляет функции:

– создания (регистрации);

– просмотра;

– редактирования;

– смены пароля;

– удаления;

– поиска;

– фильтрации;

– сортировки.

При создании нового пользователя Системы ему присваивается соответствующая роль и по умолчанию назначаются все права, доступные для пользователя указанной роли.

Для каждой роли пользователей должна быть предусмотрена возможность взаимодействия с разделом «Личный кабинет». Он обеспечивает интерфейс для просмотра пользователем Системы своих регистрационных данных, а также действий по их изменению. Пользователю доступна возможность самостоятельной смены пароля используемой им учетной записи.

### **2.3.2. Настройка прав доступа ролей**

В модуле должна быть реализована панель для настройки прав доступа каждой из ролей. Под настройкой понимается добавление, отключение и ограничение функциональных возможностей. Полный перечень функциональных возможностей приведен в пункте 3 настоящего документа.

### **2.3.3. Работа с конструктором форм отчетности**

Под работой с конструктором отчетности понимается создание, редактирование и удаление форм отчетности. Работа осуществляется через модуль “Конструктор форм отчетности”, описанный в пункте 2.6 настоящего документа.

### **2.3.4. Просмотр истории действий пользователей**

Необходимо отображать пользователю все действия, которые были совершены пользователями в Системе. Логирование действий, а также формат их отображения описаны в пункте 2.7 настоящего документа.

## **2.4. Модуль «Управление юристами»**

Модуль предназначен для работы с учетными записями юристов системы и должен предоставлять следующие функциональные возможности:

### **2.4.1. Управление учетными записями юристов в Системе**

Под управлением понимается работа с учетными записями пользователей (создание, редактирование, удаление), а также назначение и разграничение их прав, позволяющих выполнять те или иные функции в Системе.

Функционал модуля аналогичен представленному в модуле 2.3.1, но предназначен только для управления пользователями с ролью “Юрист”.

### **2.4.2. Просмотр истории действий в кабинете юриста**

Необходимо отображать пользователю все действия, которые были совершены им и прикрепленными к нему юристами в рамках работы с каждым из документов. Логирование действий, а также формат их отображения описаны в пункте 2.7 настоящего документа.

## **2.5. Модуль «Рабочее место юриста»**

Модуль предназначен для взаимодействия юриста с данными и заявками клиентов и должен предоставлять следующие функциональные возможности:

### **2.5.1. Внесение данных о клиенте в систему (в карточку клиента)**

Модуль должен обеспечивать возможность добавления карточек клиентов посредством специально разработанного веб-интерфейса.

Для регистрации необходимо предусмотреть форму ввода, которая должна содержать обязательные и необязательные поля, а также разделение полей на те, что заполняются вручную и те, что должны быть автоматически добавлены из ЕГРЮЛ / ЕГРИП посредством интеграции. Интеграция описана в пункте 5.2 настоящего документа.

Поля карточки приведены в Таблице 2.

ИНН пользователя должен заполняться автоматически после заполнения его паспортных данных. Если Системе не удается заполнить ИНН автоматически, пользователю должна отобразиться подсказка о том, что он может проверить паспортные данные на корректность ввода, либо ввести сведения об ИНН вручную.

Для случаев, когда карточка клиента была ранее создана в системе той же юридической компанией (совпадает электронная почта и / или телефон), должно отображаться оповещение. Для случаев, когда карточка была ранее создана другой юридической компанией, должна быть возможность просмотра и дополнения данной карточки, но уже с другим ID и без отображения персональных сведений контактного лица, а также сведений о связанных задачах.

Для карточки клиента должна быть реализована возможность прикрепления документов.

Таблица 2. Карточка клиента

| **Наименование поля** | **Формат поля** | **Обязательно / не обязательно** | **Только для ЮЛ** *(не применяется для ИП)* |
| --- | --- | --- | --- |
| **Сведения о контактном лице со стороны клиента** | | | |
| ФИО контактного лица | Текстовое поле | Обязательно | Нет |
| Электронная почта контактного лица | Текстовое поле | Обязательно если не введен телефон | Нет |
| Номер телефона контактного лица | Номер телефона | Обязательно если не введена почта | Нет |
| Серия и номер паспорта | Текстовое поле | Не обязательно | Нет |
| Кем выдан паспорт | Текстовое поле | Не обязательно | Нет |
| Дата выдачи паспорта | Не обязательно | Не обязательно | Нет |
| СНИЛС | Не обязательно | Не обязательно | Нет |
| ИНН клиента | Интеграция с Dadata либо Текстовое поле | Не обязательно | Нет |
| **Сведения о связанных организациях** | | | |
| Полное наименование организации | Интеграция с Dadata | Не обязательно | Нет |
| Краткое наименование организации | Интеграция с Dadata | Не обязательно | Да |
| ОГРН | Интеграция с Dadata | Не обязательно | Нет |
| Адрес регистрации | Интеграция с Dadata | Не обязательно | Нет |
| Дата регистрации | Интеграция с Dadata | Не обязательно | Нет |
| ФИО руководителя | Интеграция с Dadata | Не обязательно | Да |
| Должность руководителя | Интеграция с Dadata | Не обязательно | Да |
| ИНН руководителя | Интеграция с Dadata | Не обязательно | Да |
| Уставный капитал | Интеграция с Dadata | Не обязательно | Да |
| Сведения об основном ОКВЭД2 | Интеграция с Dadata | Не обязательно | Нет |
| Сведения о дополнительных ОКВЭД2 | Интеграция с Dadata | Не обязательно | Нет |

### **2.5.2. Создание задачи**

После заключения договора на оказание услуги, пользователь создает в Системе задачу. Все задачи, с которыми работает пользователь, должны отображаться у него в личном кабинете в формате реестра.

Модуль должен обеспечивать возможность добавления задач посредством специально разработанного веб-интерфейса.

При создании задачи у пользователя должна быть возможность выбора направления задачи:

– Работа с документацией (регистрационные действия);

– Создание заявки на выпуск ЭП.

Для добавления задачи необходимо предусмотреть форму ввода, которая должна содержать поля, приведенные в таблице 3.

Таблица 3. Поля заявки

| **Наименование поля** | **Формат поля** | **Обязательно / не обязательно** |
| --- | --- | --- |
| Наименование клиента | Выбор из выпадающего списка | Обязательно |
| Код услуги | Выбор из выпадающего списка | Обязательно |
| Наименование услуги | Автоматически отображается при выборе кода | Обязательно |

При создании задаче должна присваиваться дата создания, ID, а также статус. Статус должен изменяться по мере работы пользователя с задачей.

Доступные статусы:

– Создана: отображается для задач, которые созданы, но работы по ним еще не ведутся.

– В работе: отображается для задач, которые находятся в процессе.

– Готово: отображается для задач, которые завершены.

### **2.5.3. Работа с карточками клиентов**

В модуле должен содержаться перечень клиентов, привязанных к пользователю. В рамках обеспечения взаимодействия пользователя с карточками клиентов модуль предоставляет функции:

– создания;

– просмотра;

– редактирования;

– создания задачи;

– передачи в работу;

– удаления;

– поиска;

– фильтрации;

– сортировки.

В реестре должны быть доступны следующие поля:

– ФИО клиента;

– ИНН;

– телефон;

– количество связанных компаний;

– количество заявок;

– ответственный юрист.

В модуле должна быть реализована возможность перехода от реестра клиентов к карточке клиента. Карточка клиента должна состоять из следующих разделов:

– Профиль клиента. Должен содержать все доступные сведения о клиенте.

– Связанные компании. Должен содержать сведения обо всех компаниях, которые связаны с клиентом по его ИНН.

– Связанные задачи. Должен содержать все задачи, текущие и завершенные, которые связаны с клиентом.

### **2.5.4. Работа с реестром задач**

В модуле должен содержаться перечень задач, привязанных к пользователю. В рамках обеспечения взаимодействия пользователя с задачами модуль предоставляет функции:

– создания;

– просмотра;

– редактирования;

– удаления;

– поиска;

– фильтрации.

В реестре должны быть доступны следующие поля:

– наименование клиента;

– код оказываемой услуги;

– дата последнего изменения;

– статус;

– ответственный;

– функция редактирования;

– функция оплаты госпошлины;

– функция скачивания;

– функция копирования;

– функция архивирования.

Для отображения перечня задач необходимо использовать пагинацию. Руководитель юридической компании должен иметь возможность видеть все задачи, привязанные к его юристам с указанием ответственного за задачу юриста.

### **2.5.5. Работа с задачами**

Под работой понимается просмотр, редактирование, архивирование задачи. В рамках обеспечения взаимодействия пользователя с задачей модуль предоставляет функции:

– просмотра: переход к заполненным шаблонам документов по задаче;

– редактирования: работа с заполнением / корректировкой шаблонов документов по задаче;

– архивирования.

### **2.5.6. Просмотр истории ранее совершенных действий с документами**

Необходимо отображать пользователю все действия, которые были совершены им в рамках работы с каждой из задач. Логирование действий, а также формат их отображения описаны в пункте 2.7 настоящего документа.

### **2.5.7. Подписание документов ЭЦП**

Для заверения подготовленных документов клиентом перед отправкой их в ФНС должна быть реализована возможность подписания действующей ЭП на токене.

Для этого необходимо реализовать возможность подписи контейнера документов через КриптоПро. описание данного функционала приведено в пункте 5.4 настоящего документа.

### **2.5.8. Направление документов в ФНС**

Должна быть доступна возможность отправки подписанного контейнера документов в ФНС. Для этого должна быть реализована интеграция с ФНС, описанная в пункте 5.1 настоящего документа.

### **2.5.9. Направление запроса на выпуск ЭЦП в Удостоверяющий центр**

Должна быть доступна возможность направления запроса на выпуск ЭП для клиента в Удостоверяющий центр. Для этого должна быть реализована интеграция с УЦ Основание, описанная в пункте 5.3 настоящего документа.

### **2.5.10. Скачивание и распечатка документов**

Должна быть реализована возможность скачивания и распечатки подготовленных документов для предоставления их клиенту.

## **2.6. Модуль “Конструктор форм документов”**

Модуль предназначен для создания и редактирования форм документов.

Необходимо поддерживать в системе актуальность форм документов в соответствии с законодательством. Обновление существующих форм документов, а также добавление новых при необходимости осуществляется подрядчиком в рамках поддержки системы.

Модуль должен предоставлять следующие функциональные возможности:

### **2.6.1. Создание формы документа**

Под созданием понимается пошаговое добавление полей формы. В модуле должна быть доступна панель для создания элементов формы документа, а также рабочее поле, в котором пользователю будет отражаться результат создания элементов. Визуально рабочее поле должно выглядеть аналогично форме, которую после создания будет видеть пользователь.

Настройка свойств элементов, таких как плейсхолдер (текст-подсказка внутри заполняемого поля), заголовок, название и т.п. должна осуществляться в отдельном всплывающем окне.

В модуле должны быть доступны следующие инструменты для добавления элементов формы:

– добавление номера формы документа;

– добавление названия документа;

– добавление элемента (в том числе, поле, подзаголовок, разделитель и прочие необходимые для создания документа элементы);

– настройка свойств элемента (тип поля, обязательность, длина строки и прочее).

Для удобства работы с шаблоном его элементы должны подразделяться на слои, которые позволяют корректно размещать элементы. К слоям относятся:

– Контейнеры – позволяют сгруппировать поля для удобного отображения в административной панели, не влияют на визуальную часть.

– Шаги / табы – разделяют поля верхнеуровнево для удобного заполнения, при этом валидация шага зависит от валидации всех его полей.

– Группы – визуально группируют поля в блок.

– Метки – подписи к группам и полям.

– Подсказки – подписи к группам и полям, при наведении.

– Текст – различные заголовки.

– Колонки – делят содержимое на 2 и более колонок.

– Сообщения – различные информационные сообщения.

Правильное проектирование полей, с учетом различных настроек, таких как: метки, подсказки и т.п., необходимо чтобы убрать их из отдельных элементов, размещаемых в конструкторе и уменьшить сложность администрирования сконструированных форм.

В модуле должны быть доступны следующие типы полей:

– тестовое поле;

– текстовый редактор;

– число;

– номер телефона;

– выпадающий список;

– переключатель Да/Нет;

– дата;

– время;

– дата время;

– файл;

– изображение;

– поля для интеграции с dadata.

При заполнении полей пользователей должна выполняться их валидация. Поле может содержать одну или несколько валидаций из списка:

– обязательно или нет;

– по длине поля, минимальная и максимальная длина;

– по типу: текст, число, почта, телефон;

– по регулярному выражению.

При необходимости могут быть добавлены дополнительные варианты валидации. Для всех типов валидации необходимо предусмотреть стандартные сообщения. При необходимости сообщения могут быть индивидуальными для одного / нескольких типов валидации.

При создании полей должна быть учтена возможность их связи.

Некоторые поля являются обязательными только в зависимости от других полей. Необходимо предусмотреть настройку зависимостей полей или групп полей в случае изменения одного или нескольких других полей.

Например поле **B** зависит от поля **A**. Если поле **А** заполнено, то поле **B** обязательно тоже должно быть заполнено, и наоборот. Если заполнено поле **B**, то поле **C** должно скрываться и не быть обязательным. Получается несколько вложенных зависимостей.

Все связи делятся на два типа:

– Влияют на правила валидации

–Влияют на видимость

### **2.6.2. Привязка полей формы к PDF документу**

После добавления и настройки элементов шаблона должна быть возможность привязки полей шаблона к исходному PDF документу, который предполагается заполнить при помощи шаблона.

Визуально привязку полей формы необходимо отображать следующим образом (слева направо):

– набор полей, распознанных из PDF (ширина 25%)

– набор полей из конструктора, которые должны быть привязаны к полям PDF (ширина 25%)

– непосредственно PDF файл, не интерактивный (ширина 50%)

При наведении на выбранное поле должна быть реализована возможность прокрутить область с отображением PDF до нужного места и подсветить поле внутри PDF. При связывании поля из PDF и поля из конструктора, количество непривязанных полей конструктора должно уменьшаться, во избежание дублирования привязки.

### **2.6.3. Изменение формы документа**

Под изменением понимается редактирование и удаление формы документа.

## **2.7. Модуль “Логирование и уведомления”**

Модуль предназначен для генерации и хранения логов пользовательских действий в системе, а также для отображения уведомлений. Модуль должен предоставлять следующие функциональные возможности:

### **2.7.1. Автоматическое логирование действий**

В Системе должно быть реализовано автоматическое логирование действий, совершаемых пользователями в системе. Для каждого действия необходим следующий набор атрибутов:

– тип действия;

– дата;

– время;

– имя пользователя, вносящего изменение;

–имя клиента, с чьими документами ведется работа.

### **2.7.2. Просмотр истории залогированных действий**

В Системе должна быть реализована возможность просмотра истории залогированных действий с ролью пользователя.

При просмотре истории должна быть возможность прямого перехода к документу, который упоминается в уведомлении.

### **2.7.3. Просмотр уведомлений**

– Уведомления, связанные с задачами (новый клиент, смена юриста, изменение статуса задачи);

– Уведомление от УЦ о выпуске ЭП для клиента.

## **2.8. Модуль “Главная страница”**

Модуль предназначен для быстрого доступа пользователя к определенному при создании Системы функционалу и должен отображаться при входе пользователя в систему.

Для всех ролей на Главной странице должны отображаться лента событий согласно правам доступа и поисковая строка. Остальной набор информации и функционала Рабочего стола должен быть разработан в соответствии с ролью пользователя.

### **2.8.1. Набор информации для роли “Администратор”**

Для пользователей с ролью “Администратор” должны отображаться:

– переход к списку пользователей;

– переход к настройке прав доступа пользователей;

– переход к конструктору форм документов.

### **2.8.2. Набор информации для роли “Руководитель компании”**

Для пользователей с ролью “Руководитель компании” должны отображаться:

– список открытых запросов юристов, привязанных к указанному руководителю;

– статистика по запросам юристов компании (в работе / завершено) с возможностью выбора периода отображения данных.

### **2.8.3. Набор информации для роли “Юрист”**

Для пользователей с ролью “Юрист” должны отображаться:

– список открытых запросов юриста;

– статистика по запросам юристов компании (в работе / завершено) с возможностью выбора периода отображения данных.

# **3. Ролевая структура системы**

Система должна обеспечивать возможность назначения ролей для авторизованных пользователей.

Доступные роли приведены в Таблице 4:

Таблица 4. Ролевая структура

| **Функциональные возможности** | **Роль в системе** | | |
| --- | --- | --- | --- |
| **Администратор** | **Руководитель компании** | **Юрист** |
| Управление УЗ Руководителей компаний | + | – | – |
| Управление УЗ Юристов | + | + | – |
| Работа с конструктором форм отчетности | + | – | – |
| Просмотр истории операций | Всех пользователей | Только юристов, привязанных к руководителю | Только по привязанным заявкам |
| Изменение юриста, закрепленного за клиентом | + | + | – |
| Работа с заявками клиентов | + | Только клиенты юристов, привязанных к руководителю | Только заявки, привязанные к юристу |
| Создание запроса на выпуск ЭП | + | + | + |

# **4. Организация хранения данных**

Хранение документов и данных должно быть организовано на стороннем сервере.

# **5. Интеграции**

В рамках работ по созданию Системы должен быть реализован ряд интеграций с внешними системами, в том числе:

* Интеграция с ФНС
* Интеграция с Dadata
* Интеграция с УЦ “Основание”
* Подпись контейнера документов через КриптоПро
* Интеграция с myDSS

## **5.1. Интеграция с ФНС**

На данный момент информация по интеграции с ФНС отсутствует. При появлении необходимых сведений функционал будет оценен и добавлен отдельным приложением.

## **5.2. Интеграция с Dadata**

### **5.2.1 Валидация адреса**

Для нормализации вводимых в системе адресов необходимо использовать сервис dadata.ru. Во всех полях анкеты, где требуется указание адреса, необходимо обращаться к этому сервису для (1) получения подсказок адресов и (2) нормализации введенного адреса и получения дополнительной информации по нему (ФИАС коды частей адреса)

При отправке запросов к сервису Dadata необходимо передавать следующие значения:

* Content-Type: application/json - аналогично заполняется поле Accept
* ${API\_KEY} - API-ключ (находится в ЛК пользователя сервиса
* ${SECRET\_KEY} - секретный ключ (находится в ЛК пользователя сервиса)

Тело запроса необходимо передавать в кодировке UTF-8.

### **5.2.2 Подсказки по адресам**

При заполнении адресного поля необходимо отправлять запрос https://suggestions.dadata.ru/suggestions/api/4\_1/rs/suggest/address

в сервис dadata (ссылка на [документацию](https://dadata.ru/api/suggest/address/)) и передавать следующие параметры:

* query - в параметре передается вводимое пользователем значение
* count - количество возвращаемых подсказок, указывается целое число, необязательный параметр, по умолчанию значение равно 10

**Замечание**: первый запрос в сервис отправляется после 3го непустого символа, например, “мск”, “тур” и т д. Далее запрос отправляется при последующем добавлении символов - новый символ = новый запрос

В ответ на запрос приходит список адресов, соответствующих запросу в массиве suggestions

По каждому адресу отображаются следующие данные:

* source - Исходный адрес одной строкой
* result - Стандартизованный адрес одной строкой
* postal\_code - Индекс
* federal\_district - Федеральный округ
* region\_fias\_id - ФИАС-код региона
* region\_with\_type - Регион с типом
* region\_type\_full - Тип региона
* region - Регион
* area\_fias\_id - ФИАС-код района
* area\_with\_type - Район в регионе с типом
* area\_type\_full - Тип района в регионе
* area - Район в регионе
* city\_fias\_id - ФИАС-код города
* city\_with\_type - Город с типом
* city\_type\_full - Тип города
* city - Город
* city\_area - Административный округ (только для Москвы)
* city\_district\_fias\_id - ФИАС-код района города (заполняется, только если район есть в ФИАС)
* city\_district\_with\_type - Район города с типом
* city\_district\_type\_full - Тип района города
* city\_district - Район города
* settlement\_fias\_id - ФИАС-код населенного пункта
* settlement\_with\_type - Населенный пункт с типом
* settlement\_type\_full - Тип населенного пункта
* settlement - Населенный пункт
* street\_fias\_id - ФИАС-код улицы
* street\_with\_type - Улица с типом
* street\_type\_full - Тип улицы
* street - Улица
* house\_fias\_id - ФИАС-код дома
* house\_type\_full - Тип дома
* house - Дом
* block\_type\_full - Тип корпуса/строения
* block - Корпус/строение
* entrance - Подъезд
* floor - Этаж
* flat\_fias\_id - ФИАС-код квартиры
* flat\_type\_full - Тип квартиры
* flat - Квартира
* fias\_id - ФИАС-код адреса (идентификатор ФИАС)
  + ROOM.ROOMGUID — если квартира найдена в ФИАС
  + HOUSE.HOUSEGUID — если дом найден в ФИАС
  + ADDROBJ.AOGUID — в противном случае
* fias\_code - Иерархический код адреса в ФИАС (СС+РРР+ГГГ+ППП+СССС+УУУУ+ДДДД)
* fias\_level - Уровень детализации, до которого адрес найден в ФИАС
  + 0 — страна
  + 1 — регион
  + 3 — район
  + 4 — город
  + 5 — район города
  + 6 — населенный пункт
  + 7 — улица
  + 8 — дом
  + 9 — квартира
  + 65 — планировочная структура
  + 90 — доп. территория
  + 91 — улица в доп. территории
  + -1 — иностранный или пустой
* fias\_actuality\_state - Признак актуальности адреса в ФИАС
  + 0 — актуальный
  + 1-50 — переименован
  + 51 — переподчинен
  + 99 — удален
* okato - Код ОКАТО
* oktmo - Код ОКТМО
* timezone - Часовой пояс города для России, часовой пояс страны — для иностранных адресов. Если у страны несколько поясов, вернёт минимальный и максимальный через слеш: UTC+5/UTC+6
* qc\_house - Признак наличия дома в ФИАС
* qc - Код проверки адреса
* unparsed\_parts - Нераспознанная часть адреса. Для адреса «Москва, Митинская улица, 40, вход с торца» вернет «ВХОД, С, ТОРЦА»

При выборе того или иного значения из списка организаций. полученная информация по ней подставляется в соответствующие поля в форм

### **5.2.3 Стандартизация адреса**

При заполнении адресного поля (без подсказок) необходимо отправлять запрос https://cleaner.dadata.ru/api/v1/clean/address

в сервис dadata (ссылка на [документацию](https://dadata.ru/api/clean/address/)) и передавать следующие параметры:

* query - в параметре передается вводимое пользователем значение (обязательное).

В ответ на запрос приходит список адресов, соответствующих запросу в массиве.

По каждому адресу отображаются следующие данные:

* source - Исходный адрес одной строкой
* result - Стандартизованный адрес одной строкой
* postal\_code - Индекс
* federal\_district - Федеральный округ
* region\_fias\_id - ФИАС-код региона
* region\_with\_type - Регион с типом
* region\_type\_full - Тип региона
* region - Регион
* area\_fias\_id - ФИАС-код района
* area\_with\_type - Район в регионе с типом
* area\_type\_full - Тип района в регионе
* area - Район в регионе
* city\_fias\_id - ФИАС-код города
* city\_with\_type - Город с типом
* city\_type\_full - Тип города
* city - Город
* city\_area - Административный округ (только для Москвы)
* city\_district\_fias\_id - ФИАС-код района города (заполняется, только если район есть в ФИАС)
* city\_district\_with\_type - Район города с типом
* city\_district\_type\_full - Тип района города
* city\_district - Район города
* settlement\_fias\_id - ФИАС-код населенного пункта
* settlement\_with\_type - Населенный пункт с типом
* settlement\_type\_full - Тип населенного пункта
* settlement - Населенный пункт
* street\_fias\_id - ФИАС-код улицы
* street\_with\_type - Улица с типом
* street\_type\_full - Тип улицы
* street - Улица
* house\_fias\_id - ФИАС-код дома
* house\_type\_full - Тип дома
* house - Дом
* block\_type\_full - Тип корпуса/строения
* block - Корпус/строение
* entrance - Подъезд
* floor - Этаж
* flat\_fias\_id - ФИАС-код квартиры
* flat\_type\_full - Тип квартиры
* flat - Квартира
* fias\_id - ФИАС-код адреса (идентификатор ФИАС)
  + ROOM.ROOMGUID — если квартира найдена в ФИАС
  + HOUSE.HOUSEGUID — если дом найден в ФИАС
  + ADDROBJ.AOGUID — в противном случае
* fias\_code - Иерархический код адреса в ФИАС (СС+РРР+ГГГ+ППП+СССС+УУУУ+ДДДД)
* fias\_level - Уровень детализации, до которого адрес найден в ФИАС
  + 0 — страна
  + 1 — регион
  + 3 — район
  + 4 — город
  + 5 — район города
  + 6 — населенный пункт
  + 7 — улица
  + 8 — дом
  + 9 — квартира
  + 65 — планировочная структура
  + 90 — доп. территория
  + 91 — улица в доп. территории
  + -1 — иностранный или пустой
* fias\_actuality\_state - Признак актуальности адреса в ФИАС
  + 0 — актуальный
  + 1-50 — переименован
  + 51 — переподчинен
  + 99 — удален
* okato - Код ОКАТО
* oktmo - Код ОКТМО
* timezone - Часовой пояс города для России, часовой пояс страны — для иностранных адресов. Если у страны несколько поясов, вернёт минимальный и максимальный через слеш: UTC+5/UTC+6
* qc\_house - Признак наличия дома в ФИАС
* qc - Код проверки адреса
* unparsed\_parts - Нераспознанная часть адреса. Для адреса «Москва, Митинская улица, 40, вход с торца» вернет «ВХОД, С, ТОРЦА»

При выборе того или иного значения из списка организаций. полученная информация по ней подставляется в соответствующие поля в форме (**наверное пока рано описывать какие поля куда?**).

### **5.2.4 Поиск компании по ИНН/ОГРН**

Для заполнения данных о зарегистрированных компаниях в формах об изменении данных ЮЛ и ИП, а также создании кабинета Руководителя компании необходимо запрашивать информацию об организации по идентификатору - ИНН/ОГРН. Для этого необходимо отправлять запрос https://suggestions.dadata.ru/suggestions/api/4\_1/rs/findById/party

в сервисе dadata (ссылка на [документацию](https://dadata.ru/api/find-party/)) и передавать следующие поля:

* query - ИНН или ОГРН искомой организации (обязательное)
* branch\_type - Головная организация (передается значение MAIN) или филиал (передается значение BRANCH)

В ответ на запрос приходит список организаций, удовлетворяющих заданным параметрам. По каждой организации отображаются следующие данные

* value - Наименование компании
* data.inn - ИНН
* data.kpp - КПП
* data.ogrn - ОГРН
* data.ogrn\_date - Дата выдачи ОГРН
* data.type - Тип организации:
  + LEGAL — юридическое лицо
  + INDIVIDUAL — индивидуальный предприниматель
* data.name Наименование
  + full\_with\_opf — полное наименование
  + short\_with\_opf — краткое наименование
  + full — полное наименование без ОПФ, генерируется на основе full\_with\_opf, может содержать ошибки
  + short — краткое наименование без ОПФ, генерируется на основе short\_with\_opf, может содержать ошибки
* data.fio - ФИО индивидуального предпринимателя
  + surname — фамилия
  + name — имя
  + patronymic — отчество
* data.okato - Код ОКАТО
* data.oktmo - Код ОКТМО
* data.okpo - Код ОКПО
* data.okogu - Код ОКОГУ
* data.okfs - Код ОКФС
* data.okved - Код ОКВЭД
* data.okved\_type - Версия справочника ОКВЭД (2001 или 2014)
* data.opf - Организационно-правовая форма
  + code — код ОКОПФ
  + full — полное название ОПФ
  + short — краткое название ОПФ
  + type — версия справочника ОКОПФ (99, 2012 или 2014)
* data.management - Руководитель
  + name — ФИО руководителя
  + post — должность руководителя
* data.branch\_count - Количество филиалов
* data.branch\_type - Тип подразделения
  + MAIN — головная организация
  + BRANCH — филиал
* data.address - Адрес
  + value — адрес одной строкой: адрес организации для юридических лиц; город проживания для индивидуальных предпринимателей; стандартизован, поэтому может отличаться от записанного в ЕГРЮЛ.
  + unrestricted\_value — адрес одной строкой (полный, с индексом) стандартизован, поэтому может отличаться от записанного в ЕГРЮЛ.
  + data — гранулярный адрес
  + data.source — адрес одной строкой как в ЕГРЮЛ
  + data.qc — код проверки адреса
    - 0 — адрес распознан уверенно
    - 1 или 3 — требуется ручная проверка
* data.state - Состояние
  + actuality\_date — дата последних изменений
  + registration\_date — дата регистрации
  + liquidation\_date — дата ликвидации
  + status — статус организации
    - ACTIVE — действующая
    - LIQUIDATING — ликвидируется
    - LIQUIDATED — ликвидирована
    - BANKRUPT — банкротство
    - REORGANIZING — в процессе присоединения к другому юрлицу, с последующей ликвидацией
  + code — детальный статус (только на «Максимальном» тарифе)
* data.okved - Код ОКВЭД
* data.okveds[ ] - Коды ОКВЭД дополнительных видов деятельности
  + main — основной или нет (true / false)
  + type — версия справочника ОКВЭД (2001 или 2014)
  + code — код по справочнику
  + name — наименование по справочнику
* data.authorities - Сведения о налоговой, ПФР и ФСС
* data.authorities.fts\_registration - ИФНС регистрации
  + type — код гос. органа
  + code — код отделения
  + name — наименование отделения
  + address — адрес отделения одной строкой
* data.authorities.fts\_report - ИФНС отчётности, структура аналогична fts\_registration
* data.authorities.pf - Отделение Пенсионного фонда, структура аналогична fts\_registration
* data.authorities.sif - Отделение Фонда соц. страхования, структура аналогична fts\_registration
* data.citizenship - Гражданство ИП
  + code.numeric — числовой код страны по ОКСМ
  + code.alpha\_3 — трехбуквенный код страны по ОКСМ
  + name.full — полное наименование страны
  + name.short — краткое наименование страны
* data.founders[ ] - Учредители компании
  + ogrn — ОГРН учредителя (для юрлиц)
  + inn — ИНН учредителя
  + name — наименование учредителя (для юрлиц)
  + fio — ФИО учредителя (для физлиц)
  + hid — внутренний идентификатор
  + type — тип учредителя (LEGAL / PHYSICAL)
* data.founders[ ].share - Доля учредителя
  + type — тип значения (PERCENT / DECIMAL / FRACTION)
  + value — значение (для type = PERCENT и type = DECIMAL)
  + numerator — числитель дроби (для type = FRACTION)
  + denominator — знаменатель дроби (для type = FRACTION)
* data.management.disqualified - Наличие дисквалифицированных лиц в руководстве компании. Возвращает true при наличии, иначе null.
* data.managers[ ] - Руководители компании
  + ogrn — ОГРН руководителя (для юрлиц)
  + inn — ИНН руководителя
  + name — наименование руководителя (для юрлиц)
  + fio — ФИО руководителя (для физлиц)
  + post — должность руководителя (для физлиц)
  + hid — внутренний идентификатор
  + type — тип руководителя
    - EMPLOYEE — сотрудник
    - FOREIGNER — иностранный гражданин
    - LEGAL — юрлицо
* data.predecessors[ ] - Правопредшественники, только для юрлиц
  + ogrn — ОГРН предшественника
  + inn — ИНН предшественника
  + name — наименование предшественника
* data.successors[ ] - Правопреемники, только для юрлиц
  + ogrn — ОГРН преемника
  + inn — ИНН преемника
  + name — наименование преемника
* data.capital - Уставной капитал компании
  + type — тип капитала
  + value — размер капитала
* data.documents - Документы
* data.documents.fts\_registration - Свидетельство о регистрации в налоговой
  + type — тип документа (= FTS\_REGISTRATION)
  + series — серия документа
  + number — номер документа
  + issue\_date — дата выдачи
  + issue\_authority — код подразделения
* data.documents.fts\_report - Сведения об учете в налоговом органе, структура аналогична fts\_registration
* data.documents.pf\_registration - Свидетельство о регистрации в Пенсионном фонде, структура аналогична fts\_registration
* data.documents.sif\_registration - Свидетельство о регистрации в Фонде соц. страхования, структура аналогична fts\_registration
* data.documents.smb - Запись в реестре малого и среднего предпринимательства
  + type — тип документа (= SMB)
  + category — категория предприятия (MICRO, SMALL или MEDIUM)
  + issue\_date — дата регистрации в реестре

При выборе того или иного значения из списка организаций. полученная информация по ней подставляется в соответствующие поля в форме (**наверное пока рано описывать какие поля куда?**).

## **5.3. Интеграция с УЦ “Основание”**

Для реализация процесса создания ЭЦП для клиентов нужна интеграция с удостоверяющим центром “Основание”. Процесс выдачи будет состоять из следующих шагов

Для того, чтобы обращаться в API УЦ “Основание” необходимо получить данные для авторизации (логин и пароль), чтобы в *каждом* методе при обращении передавать иж в полях "login" и "pass" соответственно.

### **5.3.1 Создание заявки**

Для создания заявки в УЦ “Основание” необходимо передавать метод <https://apinew.iecp.ru/api/external/v2/request/create> cо следующими полями (набор полей зависит от типа заявки - физическое лицо (type = 1), ИП (type = 2) или ЮЛ (type = 3)):

| **Название поля** | **Описание** | **Обязательно для ФЛ** | **Обязательно для ИП** | **Обязательно для ЮЛ** |
| --- | --- | --- | --- | --- |
| firstName | Имя заявителя | + | + | + |
| middleName | Отчество заявителя | + | + | + |
| lastName | Фамилия заявителя | + | + | + |
| company | Наименование организации | + | + | + |
| position | Название должности | + | + | + |
| department | Название департамента | + | + | + |
| passportSerial | Серия паспорта | + | + | + |
| passportNumber | Номер паспорта | + | + | + |
| passportDate | Дата выдачи | + | + | + |
| passportCode | Код подразделения | + | + | + |
| passportDivision | Кем выдан | + | + | + |
| gender | гендер: *M* - мужской, *F* - женский, *U* - не определен | + | + | + |
| birthDate | Дата рождения | + | + | + |
| inn | ИНН организации / ИП | + | + | + |
| personInn | ИНН заявителя | + | + | + |
| ogrn | ОГРН организации или ОГРНИП ИП | + | + | + |
| kpp | КПП организации/ИП | + | + | + |
| companyPhone | Телефон организации /ИП | - | + | + |
| headLastName | Отчество руководителя организации | - | + | + |
| headFirstName | Отчество руководителя организации | - | + | + |
| headMiddleName | Отчество руководителя организации | - | + | + |
| HeadPosition | Название позиции руководителя | - | - | + |
| regionLaw | Регион юр адреса организации | - | + | + |
| cityLaw | Город юр адреса организации | - | + | + |
| addressLaw | Адрес организации (из юридического только улица, дом, офис) | - | + | + |
| index | Почтовый индекс юр адреса организации | - | + | + |
| snils | СНИЛС | + | - | - |
| email | электронная почта | + | + | + |
| phone | номер телефона | + | + | + |
| region | идентификатор региона, передается официальный код региона см [тут](https://hramy.ru/regions/regfull.htm) | + | + | + |
| city | населенный пункт | + | + | + |
| address | адрес | + | + | + |
| type | тип заявителя (1 - физ. лицо, 2 - ИП, 3 - юр. лицо) | + | + | + |
| products | идентификаторы продуктов, которые ассоциируются с заявкой (см [метод получения списка продуктов](#q4z6kc2j55pv)) | - | - | - |
| offerJoining | если у клиента есть присоединение к оферте - true, иначе false (по-умолчанию false). В  случае если присоединения к оферте нет, то ожидается, что будет скан загружен скан договора  оферты в архиве документов. | + | + | + |
| identificationKind | Способ идентификации: 0 - Лично, 1 - По действующей ЭП.  Если не указано, будет передано 0 | - | - | - |

В ответ на успешный запрос приходит идентификатор запроса - *requestId.* В дальнейшем по нему можно будет производить остальные операции

### **5.3.2 Изменение данных по заявке**

Для изменения информации по заявкам нужно отправить POST запрос на адрес: <https://apinew.iecp.ru/api/external/v2/request/change> Параметры запроса аналогичны методу создания заявки (см [Создание заявки](#h3u3r1157f1o)) и добавляется новое обязательное поле - *requestId* - идентификатор заявки, который был получен при ее создании.

### **5.3.3 Перемещение заяви в архив**

Для переноса заявки в архив необходимо отправить POST запрос на адрес: <https://apinew.iecp.ru/api/external/v2/request/change>, в котором передавать *requestId* - идентификатор заявки, который был получен при ее создании и параметр *requestArchived* со значением true

### **5.3.4 Просмотр сведений о существующей заявке**

Для просмотра сведений о существующей заявке нужно отправить POST запрос на адрес <https://apinew.iecp.ru/api/external/v2/request/view> , в которой необходимо передавать *requestId* исковой заявки

В ответ вернется вся информация по заявки:

* Информация по заявке - тот же формат, что и при ее [создании](#h3u3r1157f1o)
* Статус заявки:
  + statusId - идентификатор статуса заявки
  + status - расшифровка статуса заявки
* Информация list

### **5.3.5 Прикрепление документов к существующей заявке**

Для корректного прохождения жизненного цикла необходимо прикрепить следующие файлы в следующей последовательности:

1. Файл zip архива с документами (указывается тип файла 8), должны содержать следующие файлы:
   1. Доверенность на представителя (скан) (обязательно для ЮР лиц)
   2. Паспорт заявителя (скан) (обязательно для всех заявок)
   3. СНИЛС заявителя (скан) (обязательно для всех заявок)
   4. Фото заявителя (обязательно для всех заявок)
   5. Заявление на выпуск СКП ЭП (скан) (обязательно для всех заявок)
   6. Клиентский договор на оказание услуг по выпуску СКП ЭП (скан) (обязательно при выборе значения "Нет" в поле "Присоединение к оферте" на вкладке "Форма заявки") При загрузке файла есть проверка на количество файлов внутри архива: Для юр лиц - минимум 6 файлов. Для остальных - минимум 5 файлов При прикреплении файла архива статус заявки не меняется.
2. Файл подписи zip архива из п.1 в формате ГОСТ-3411, ГОСТ-3411 2012 256 бит или ГОСТ-3411 2012 512 бит. Указывается тип файла 9 При прикреплении этого файла заявка переходит в статус Генерация запроса
3. Файл запроса на сертификат в формате p10. Указывается тип файла 0 При прикреплении этого файла заявка переходит в статус Выпуск

Для добавления файлов в заявку нужно отправить POST запрос на адрес <https://apinew.iecp.ru/api/external/v2/request/attach_file> со следующими полями:

* fileName - название фала
* file - содержимое файла закодированное в base64
* fileType - тип файла, может принимать следующие значения:
  + 0 - загружается файл запроса на сертификат, после этого заявка переводится в статус Выпуск
  + 8 - загружается файл архив с документами (сканы документов заявителя, подписанные заявления), заявка статус не меняет
  + 9 - загружается файл .sig с подписью архива документов, заявка переводится в статус\_ Генераци запроса\_
* requestId - идентификатор заявки, к которой необходимо прикрепить документы

В ответ в случае успешного прикрепления файла приходит 200 ОК с пустым телом.

### **5.3.6 Получение списка продуктов**

Для получения списка продуктов, ассоциированных с организацией менеджера, логин которого указан в параметре Login необходимо отправить запрос на адрес <https://apinew.iecp.ru/api/external/v2/products> Никаких дополнительных параметров кроме авторизации данный метод не требует

В ответ возвращается список продуктов в массиве products, где по каждому продукту есть следующая информация:

* id - идентификатор продукта
* name - название продукта
* price - стоимость продукта для ФЛ, ИП или ЮЛ передается в соответствующих полях "fl", "ip" и "ur"
* oids
* required - принимает значение true/false, обозначает обязательный продукт или продукт по умолчанию, который обязательно должен присутствовать в запросе на сертификат.

## **5.4. Подпись контейнера документов через КриптоПро**

Для возможности подписи документов ЭЦП необходимо настроить работу с сервисом КриптоПро. На текущий момент подпись документов будет происходить с помощью подключения ЭЦП-носителя (флеш-карты) к рабочему компьютеру юриста.

### **5.4.1 Подготовка к подписанию документа**

Перед началом использования ЭЦП пользователь-юрист должен убедиться, что у него на ПК имеются все необходимые инструменты и средства. К ним относят:

* криптопровайдер - установка программы КриптоПро CSP (см [информацию](https://www.cryptopro.ru/products/cades/plugin)) для использования электронной подписи;
* закрытый ключ и сертификат ЭЦП (в данном случае принесенный пользователем );
* настроенное рабочее место.

Для того, чтобы иметь возможность подписать документ любого формата, необходимо установить программный модуль КриптоАРМ (см. [информацию](https://www.cryptopro.ru/products/partner/crypto-arm)). Затем выполнить следующие шаги:

* выбираем в меню пункт “Подписать”;
* нажимаем на «Выбор файла», переходим к файлу, который нужно подписать. Щелкаем по нему и жмем «Далее»;
* появляется окошко «Выходной формат». Если нет обязательных требований к типу кодировок, ничего не меняем. Кроме этого есть возможность сохранить в формате ZIP, и выбрать место для сохранения результатов подписи. Жмем «Далее»;
* выбираем формат подписи - существуют два вида, присоединенная и отсоединенная. В первом случае с присоединенной, ЭП будет встроена к подписываемому документу. Во втором случае с отсоединенной подписью, подписываемый документ остается неизменным, и подпись создается в отдельном файле с расширением \*.sig. Когда все готово, жмем «Далее».
* выбираем сертификат подписи, которым нужно подписать документ, и нажимаем «Далее»;
* видим итоговое окно с общей информацией. Обращаем внимание на опцию «Сохранить данные в профиль для дальнейшего использования» и нажимаем сохранить профиль. Нажимаем на кнопку «Готово». Система по окончании установки покажет сообщение, об успешной установке, и что выбранный файл подписан.

## **5.5. Интеграция с myDSS**

При появлении возможности и необходимости может быть добавлена интеграция с myDSS. Описание интеграции приведено в отдельном [документе](https://docs.google.com/document/d/1V28dPLdvxd7G2I7N8TJ3hxwpT2gvUyeF_p-em3qRVzs/edit). Оценка добавления интеграции будет выполнено отдельно в данном случае.

# **6. Схема бизнес-процессов**

Схема бизнес-процессов размещена в Приложении 1 к настоящему документу. На ней отображены цепочки действий пользователей со всеми типами ролей, разделение процессов друг от друга.