**ТЗ по API сервису (MVP)**

**1. Раздел с общей информацией**

**1.1. Главный экран сервиса**

Данный экран является основной точкой входа на uсервис. Переход на этот экран не валидируется на наличие авторизации.

Хэдер и футер, которые присутствуют на данном экране будут также отображаться на всех других экранах сервиса (кроме экранов раздела администратора).

Хэдер будет содержать логотип и название сервиса (если предусмотрено дизайном) и блок управления авторизацией. Содержимое этого блока будет зависеть от того, авторизован пользователь или нет. Если нет - там будет форма логина, которая состоит из поля e-mail и поля для ввода пароля и кнопки “Авторизоваться”. Валидация этой формы осуществляется на стороне сервера.

Если валидация не пройдена, возле формы отображается соответствующее сообщение об ошибке. Если валидация пройдена успешно, то авторизованный пользователь перенаправляется или на главный экран области пользователя-юнита, или на главный экран пользователя-агента, или на главный экран раздела администратора (в зависимости от роли пользователя).

Также в блоке управления авторизацией (для неавторизованного пользователя) будут отображаться кнопки: “Забыл пароль”, которая перенаправляет на экран восстановления пароля, и “Зарегистрироваться”, которая перенаправляет на экран регистрации.

Для авторизованного пользователя в этом блоке будет отображаться кнопка “Выйти”, которая разлогинивает пользователя, а также информация о e-mail авторизованного в данный момент пользователя и его роли.

Футер будет содержать стандартный блок с текущим годом и копирайтом, также там будет ссылка на e-mail для техподдержки (с пояснением), ссылка на экран с legal terms и краткие контактные данные.

В основной части экрана будет лендингоподобное краткое описание возможностей сервиса, описание доступных типов обработчиков данных (определенный тип обработчика данных может объединять в себе несколько типов воркеров - например, FaceDetector и BboxClustering, но эта информация будет скрыта от пользователя) с соответствующими ссылками на демки и разделы с техническим описанием для разработчиков.

Также будет присутствовать панель навигации с такими ссылками: “Главная страница” - активна, соответствует текущему экрану, “Демо API” - перенаправляет на главный экран раздела демонстрации работы API, “Информация для разработчиков” - перенаправляет на главный экран раздела информации для разработчиков, “Стоимость услуг” - перенаправляет на экран прайса, “Личный кабинет” (кнопка доступна только уже авторизованным пользователям, но не пользователю-администратору) - перенаправляет на главный экран области пользователя с соответствующей ролью (главный экран пользователя-агента или главный экран пользователя-юнита).

**1.2. Экран прайса**

Переход на этот экран не валидируется на наличие авторизации.

На данном экране будет отображаться краткое описание как и с какими платежными системами мы работаем (на данный момент это Stripe и PayPal). Также будет отдельный блок с информацией о доступных типах обработчиков данных и о стоимости и условиях их использования.

На этом экране будет присутствовать такая же панель навигации, как и на главном экране сервиса с соответствующей активной кнопкой.

**1.3. Экран сообщения об ошибке сервера**

Переход на этот экран не валидируется на наличие авторизации.

Это стандартный экран ошибки сервера, он будет отображаться для всех самых распространенных кодов ошибок. Стандартного футера и хэдера здесь не будет.

На этом экране также будет ссылка на главную страницу сервиса и ссылка на e-mail для техподдержки (с пояснением). Также тут будет наша краткая контактная информация (например, телефон и/или e-mail).

**2. Раздел информации для разработчиков**

**2.1. Главный экран раздела информации для разработчиков**

Переход на этот экран не валидируется на наличие авторизации.

Этот раздел будет содержать подробное описание процедуры подключения и использования всех видов обработчиков данных.

На главной странице будет краткое описание процедуры использования данного раздела и инструкция по использованию поиска. Также здесь будет находиться форма поиска, она будет состоять из текстового поля и кнопки “Найти”. При нажатии на эту кнопку, если поле поиска не пустое, чуть ниже будет отображаться список ссылок на экраны страниц с информацией, по которым были найдены совпадения.

Основную часть экрана будет занимать оглавление, которое будет содержать заголовки и ссылки на соответствующие экраны разделов с информацией.

Возле каждого пункта оглавления будет очень краткое описание раздела.

После обсуждения формата общения с API нужно примерно продумать структуру разделов с описаниями и добавить ее сюда.

На этом экране будет присутствовать такая же панель навигации, как и на главном экране сервиса с соответствующей активной кнопкой.

**2.2. Экран раздела с информацией**

Переход на этот экран не валидируется на наличие авторизации.

На этом экране будет подробное структурированное описание соответствующего логического раздела. Все блоки, содержащие JSON или примеры кода, должны быть отформатированы и выделены.

Также должны быть ссылки на предыдущий и следующий раздел, а также на главный экран раздела информации для разработчиков.

**3. Раздел демонстрации работы API**

**3.1. Главный экран раздела демонстрации работы API**

Переход на этот экран не валидируется на наличие авторизации.

Содержимое этого экрана будет состоять из нескольких блоков, которые будут поэтапно появляться перед пользователем (как следствие его определенных действий).

На первом этапе мы отображаем форму выбора типа обработчика данных (их полный список будет доступен после обсуждения). Форма состоит из ряда радиобаттонов для выбора конкретного обработчика с кратким описанием возможностей каждого из них. Также будет кнопка “Выбрать”.

После клика по кнопке “Выбрать” пользователь переходит на второй этап, теперь появляется блок с примерами тестовых данных (отфильтрованных в зависимости от выбранного обработчика), который будет представлять из себя набор тайлов с изображениями, которые будут соответствовать содержимому файлов-примеров (картинок, видеофайлов и т.д.). Все файлы примеры будут подготовлены заранее, загружать свои файлы пользователь не сможет.

После клика на тайле файла-примера пользователь переходит на третий этап, появляется блок с подробной информацией об исходных данных запроса к API.

Этот блок будет разделен на две части: в одной части мы отобразим файл-пример (если это картинка - просто показываем исходную картинку, если видеофайл или аудиофайл - у пользователя будет возможность запустить его во встроенном плеере), а в другой части - специальный блок с код-сниппетом (соответственно отформатированный), который содержит пример кода, позволяющий обратиться к API, его содержимое соответствует типу выбранного обработчика данных и типу файла-примера. Рядом с этим код-сниппетом будет кнопка “Скопировать” - при ее нажатии содержимое сниппета копируется в буфер.

Также в блоке будет кнопка “Отправить запрос”, при нажатии на нее пользователь переходит на четвертый этап, появляется блок с подробной информацией о результате запроса к API (настоящего запроса к API в этот момент не происходит). Этот блок тоже будет разделен на две части: в одной мы покажем пример визуализации данных из ответа API - это будет один из заранее подготовленных файлов (соответствующий типу выбранного исходного файла) с визуализацией результата процессинга (соответствующей типу выбранного обработчика), а в другой части блока будет специальный блок с содержимым соответствующего ответа API в формате JSON, оформленным как код-сниппет с прокруткой. Рядом с этим код-сниппетом будут две кнопки: “Скопировать” - при ее нажатии содержимое сниппета копируется в буфер) и “Загрузить в виде файла” - при ее нажатии содержимое сниппета загружается в систему пользователя в виде JSON файла.

Таким образом, мы наглядно покажем пользователю все детали процесса отправки запроса и получения ответа от API. В блоки также будет необходимо добавить ссылки на соответствующие экраны страниц и разделов с информацией из раздела информации для разработчиков. Для каждого этапа также нужно предусмотреть краткое описание, оно должно появляться на экране вместе с соответствующими блоками.

Для этого экрана нужно предусмотреть возможность перехода на него с заранее выбранными опциями для всех этапов (указываются с помощью значений соответствующих переменных в URL экрана), чтобы при таком переходе последовательного перехода по этапам не происходило, а загружались сразу все блоки. В этом случае можно будет с помощью прямых ссылок перенаправлять пользователя на соответствующие демонстрационные примеры из раздела общей информации и технической информации.

**4. Пользовательский раздел**

**4.1 Система пользовательских ролей и доступов**

Пользователь может иметь одну из двух ролей:

ROLE\_UNIT - единственная роль, доступная при регистрации (присваивается по умолчанию). Пользователь с этой ролью указывает в своей учетной записи все необходимые legal параметры, и именно этот пользователь будет иметь доступ к статистике аккаунта (и всех аккаунтов привязанных агентов, но без явного указания их ключей), а также к разделу баланса и платежей.

При создании пользователя-юнита автоматически создается, и, в обязательном порядке, привязывается (к упомянутому пользователю-юниту) дефолтный пользователь-агент.

Только у таких дефолтных пользователей-агентов (по одному на каждый юнит) в БД поле IS\_DEFAULT будет со значением TRUE. Доступы к обработчикам данных для этого дефолтного агента пользователь-юнит сможет выставить потом на экране информации об агенте. E-mail для этого дефолтного пользователя-агента заполняется псевдо-значением, а в системе, везде, где он нужен - используется e-mail привязанного пользователя-юнита.

Дефолтный агент нужен, чтобы избавить пользователей от необходимости создания дополнительного агента (с другой почтой), если они планируют работать с API индивидуально. Дефолтный агент считается активированным с момента своего создания.

После подтверждения успешной регистрации и заполнения всех необходимых полей учетной записи, пользователь с этой ролью получает возможность рассылать приглашения на e-mail (соотв. пользователь создается в момент отправки письма) для создания сети привязанных к этой учетной записи агентов.

Пользователь с этой ролью может открывать и закрывать доступы своих агентов к определенным типам обработчиков данных. В случае предоставления доступа к одному или нескольким типам обработчиков, у пользователя-агента появляется возможность запросить создание API ключа, который может быть использован для обращения ко всем доступным для агента типам обработчиков. В случае закрытия доступа к определенному типу - все ключи API этого агента, которые содержат доступ к этому типу, сразу деактивируются.

Также пользователь-юнит может полностью заблокировать своего пользователя-агента (за исключением своего дефолтного агента, его нельзя заблокировать). В этом случае учетная запись агента деактивируется (он больше не сможет авторизоваться), а все его ключи API блокируются.

Пользователю-юниту доступны истории создания, удаления, обновления ключей API всех своих агентов (без явного указания этих ключей).

Пользователь-юнит не сможет отправлять приглашения на e-mail, которые уже зарегистрированы в системе.

При любых манипуляциях пользователя-юнита с пользователем-агентом - последнему отправляется соответствующее уведомление на e-mail (за исключением дефолтных агентов, у которых IS\_DEFAULT = TRUE).

ROLE\_AGENT - роль, которую можно получить только по приглашению пользователя-юнита.

После получения приглашения от аккаунта юнита, агент переходит по ссылке из e-mail, потом он должен установить свой пароль, и сразу может получить доступ в систему. В этот момент пользователь-юнит получает уведомление на e-mail о том, что его агент активировал свою учетную запись.

Пользователь-агент имеет доступ к своему списку API ключей. В любой момент пользователь-агент может обновить любой свой API ключ - старый вариант блокируется, выдается новый (по нему можно обращаться ко всем доступным агенту типам обработчиков данных). Также он может заблокировать любой ключ или запросить дополнительный.

Все манипуляции с ключами со стороны агентов будут сохранятся в отдельную таблицу в БД, и список с историей этих манипуляций доступен пользователю-агенту (с явным указанием ключей) и соответствующему пользователю-юниту (без явного указания ключей).

Также пользователь-агент будет иметь доступ к статистике использования всех своих ключей (с явным указанием).

**4.2. Экран регистрации**

На экране регистрации находится форма регистрации, одним из обязательных полей которой является капча (можно взять стандартную гугл-капчу).

Список полей формы регистрации для заполнения информации о пользователе (все обязательны для заполнения): e-mail пользователя, пароль (проверяется на обязательное наличие разных регистров, цифр), дублируется полем повторного ввода пароля, поле ввода цели использования сервиса (выпадающий список из предопределенных вариантов и возможность ввести свой вариант), страна, город, выбор из трех радиобаттонов: физическое лицо, юридическое лицо, научная организация. В случае выбора физического лица - появляются поля: должность, название организации, а в случае выбора научной организации или юридического лица - название.

В эту форму также будет входить чекбокс “Free category account” (с контекстной подсказкой с описанием, что это такое). В Legal Terms сервиса будут перечислены категории пользователей, которые смогут пользоваться сервисом бесплатно. Если при регистрации пользователь активирует этот чекбокс (его поле в БД REQUESTED\_FREE будет TRUE), то мы потом в ручном порядке проверяем соответствие условиям, и через админку присваиваем ему статус бесплатного пользователя (меняем значение поля FREE\_STATUS с PENDING на ACCEPTED или DECLINED).

Также необходим обязательный чекбокс с подтверждением ознакомления с Legal Terms (со встроенной ссылкой на экран Legal Terms, открывается в новой вкладке).

Под формой регистрации находится кнопка «отправить», которая неактивна, пока все необходимые поля не будут заполнены.

Все поля формы обязательно валидируются именно на стороне сервера, дефолтная фронтовая валидация (стандартная браузерная) будет отключена.

Поля дат и географических названий должны быть выполнены в форме селекторов значений. Проверка всех полей на заполненность проходит на фронте по событию изменения значения внутри любого поля, как только все поля заполнены — активируется кнопка «отправить».

После отправки данных происходит валидация на стороне сервера, в случае обнаружения ошибок, пользователь перенаправляется обратно на экран регистрации, на котором под соответствующими полями отображаются соответствующие сообщения об ошибках, при этом все заполненные поля остаются с теми же значениями.

При успешном прохождении валидации в БД создается новый пользователь с ролью ROLE\_UNIT с полем IS\_ACTIVATED со значением FALSE.

Затем сразу автоматически создается пользователь с ролью ROLE\_AGENT с полем IS\_ACTIVATED со значением TRUE и с полем IS\_DEFAULT со значением TRUE, также этот пользователь привязывается к только что созданному пользователю-юниту.

К этому дефолтному агенту автоматически привязывается демо-ключ (подробное описание работы демо-ключа в разделе модуля работы с платежами).

Доступы к обработчикам данных для этого дефолтного агента пользователь-юнит сможет выставить потом на экране информации об агенте. E-mail для этого дефолтного пользователя-агента заполняется псевдо-значением, а в системе, везде, где он нужен - используется e-mail привязанного пользователя-юнита. Пароль для этого дефолтного пользователя-агента в БД выставляется такой же, что и для привязанного к нему юнита. Залогиниться именно дефолтным пользователем-агентом не получится, только родительским юнитом.

Пользователю (на указанный при регистрации e-mail) уходит письмо со ссылкой подтверждения.

После этого пользователь перенаправляется на экран сообщения об успешной регистрации.

Когда пользователь перейдет по ссылке активации из письма, его поле IS\_ACTIVATED меняет значение на TRUE. После этого пользователь перенаправляется на экран сообщения об успешной активации пользователя-юнита.

**4.2.1. Экран Legal Terms**

Переход на этот экран не валидируется на наличие авторизации.

Этот экран всегда будет открываться в новой вкладке. На данном экране будет отображаться отформатированный и структурированный текст Legal Terms нашего проекта с указанием последней даты обновления текста.

**4.3. Экран сообщения об успешной регистрации**

Переход на этот экран обязательно валидируется на наличие авторизации.

Пока пользователь не активирует свой e-mail, это единственный экран, который будет ему доступен после авторизации.

На этом экране мы сообщаем пользователю о том, что регистрация прошла успешно, и что он теперь должен перейти по ссылке активации из e-mail. Также тут будет находиться кнопка «повторно отправить письмо активации», которая, при нажатии, повторно отправляет соответствующее сообщение на e-mail. Повторная отправка возможна 1 раз в 30 минут, после нажатия на нее рядом появится строка с информацией о том, сколько остается времени до следующей отправки. Соответственно, в БД для пользователя сохраняем дату и время последней отправки письма со ссылкой активации. (Возможно, также следует добавить капчу — обсудить.)

**4.4. Экран сообщения об успешной активации пользователя-юнита**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_UNIT и на значение поля IS\_ACTIVATED, которое должно быть TRUE.

На этом экране мы сообщаем пользователю, что активация аккаунта прошла успешно, и вкратце описываем возможности работы с системой (обязательно сообщаем, что теперь он должен добавить в систему своих пользователей-агентов или сразу воспользоваться возможностями своего привязанного дефолтного пользователя-агента). Также рассказываем, на всякий случай, как связаться с техподдержкой. Также на экране есть кнопка «ОК», при нажатии на которую пользователь перенаправляется на главный экран области пользователя-юнита.

**4.5. Главный экран области пользователя-юнита //+**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_UNIT и на значение поля IS\_ACTIVATED, которое должно быть TRUE.

К этому экрану имеет доступ только авторизованный пользователь-юнит. На этот экран можно перейти после активации аккаунта пользователя-юнита или по кнопке «Личный кабинет» (она будет доступна только авторизованным пользователям) с главного экрана сервиса.

На этом экране доступно несколько вкладок (можно оформить просто как кнопки): управление агентами (вкладка активна по умолчанию), управление данными юнита, управление платежами, управление ключами (нажатие на эту вкладку перенаправляет на главный экран области пользователя-агента, который формируется на основе данных привязанного к текущему юниту дефолтного пользователя-агента).

**4.5.1. Вкладка управления агентами //+**

На этой вкладке будет находиться кнопка «Добавить агента» (если платежные данные соответствующего пользователя-юнита не были заполнены и подтверждены, то кнопка недоступна (для подтвержденных пользователей бесплатной категории доступна), см. описание раздела модуля работы с платежами), при нажатии на нее пользователь перенаправляется на экран добавления агента.

Также здесь находится список привязанных к данному юниту агентов (изначально будет отображаться только дефолтный агент, других можно добавить позже). Обязательно присутствует пагинация.

Также есть сортировка по e-mail, дате приглашения агента, общему количеству активных ключей агента. Список можно фильтровать по строке (или части строки) e-mail, а также по набору типов обработчиков данных, к которым у агента есть доступ.

В списке отображается e-mail агента (в случае дефолтного агента вместо e-mail - просто строка “My default agent”), дата и время приглашения, статус активации аккаунта агента, список типов обработчиков данных (в виде плашек), к которым у него есть доступ, общее число активных ключей. При нажатии на e-mail агента, пользователь перенаправляется на экран информации об агенте.

Также на этой вкладке будет сводная статистика по всем агентам (можно добавить пару графиков): общее число агентов, общее количество активных ключей, общее число запросов по ключам на этот момент (на пайчартах можно отображать эти показатели по типам обработчиков данных).

**4.5.1.1. Экран добавления агента //+**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_UNIT и на значение поля IS\_ACTIVATED, которое должно быть TRUE.

На этом экране находится форма добавления агента. Она состоит из полей e-mail и чекбоксов выбора одного или нескольких типов обработчиков данных, к которым нужно дать доступ агенту.

Форма полностью валидируется на стороне бэка, если поле e-mail не проходит валидацию, то пользователь перенаправляется обратно на форму, где отображается соответствующая ошибка. Один из элементов валидации — пользователь-юнит не может отправить приглашение по адресу, который уже зарегистрирован в системе. +

Также есть кнопка «пригласить» (не активна, пока не заполнено поле e-mail) - после нажатия на нее (если валидация прошла успешно), агенту на указанный e-mail приходит письмо с объяснением (стандартным) и со ссылкой на активацию аккаунта, которая перенаправляет на экран активации аккаунта пользователя-агента. В этот момент в БД создается новый пользователь-агент, а пользователь-юнит перенаправляется обратно на вкладку управления агентами, где в списке уже появляется только что добавленный агент (со статусом активации аккаунта FALSE).

**4.5.1.2. Экран информации об агенте //+**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_UNIT и на значение поля IS\_ACTIVATED, которое должно быть TRUE. Также обязательна проверка на наличие привязки этого агента к текущему авторизованному юниту.

На этом экране будет модуль основной информации об агенте, он будет содержать: e-mail агента (для дефолтного агента - строку “My default agent”), дату и время приглашения, дату и время активации аккаунта агента, статус активации аккаунта агента, статус блокировки агента, список доступных агенту типов обработчиков данных.

Список доступных типов обработчиков данных будет представлен в виде набора чекбоксов. Заполняя и освобождая их можно будет управлять набором доступных типов обработчиков данных для агента, это реализуется через AJAX в реальном времени. Каждое изменение подтверждается в уточняющем попапе «вы уверены?».

При добавлении доступа к новому типу обработчика, у пользователя-агента появляется возможность запросить создание API ключа, с помощью которого можно будет получить доступ ко всем доступным для агента типам обработчиков (включая новый). В случае закрытия доступа к определенному типу - все ключи API этого агента, которые содержат доступ к этому типу, сразу деактивируются.

Также здесь будет модуль управления агентом, он будет содержать кнопку блокировки агента «Блокировать» (недоступна для дефолтного агента), при ее нажатии — все активные ключи агента блокируются (средствами модуля связи с API), статус блокировки меняется на TRUE, а сам агент не может больше попасть после авторизации никуда кроме как на экран уведомления о блокировке агента. Также агенту на его почту приходит соответствующее уведомление.

Если статус блокировки агента TRUE – данная кнопка меняется на кнопку «Разблокировать». При ее нажатии блокировка с агента снимается: статус блокировки становится FALSE, агент снова может получить доступ ко всем своим экранам, также он получает соответствующее уведомление на почту. Теперь он снова сможет запросить создание ключа, по которому он сможет получить доступ ко всем типам обработчиков данных, к которым у него ранее был доступ.

Также будет кнопка «Удалить» (недоступна для дефолтного агента) - при ее нажатии все активные ключи агента блокируются, и связь между юнитом и агентом уничтожается, он больше не будет отображаться в списке (запись пользователя удаляется из БД, т. е. агента можно пригласить снова при желании). Восстановить обычным способом через интерфейс его больше не получится. Удаленный агент получает соответствующее уведомление на почту.

Все вышеперечисленные кнопки внутри модуля управления агентом обязательно вызывают попап с подтверждением «вы уверены?». Для кнопки «удалить» нужно добавить сообщение, что данное действие необратимо.

Также на этом экране будет модуль информации о ключах агента, он будет содержать две главных части: часть с полной статистикой по ключам: общее кол-во активных ключей по типам обработчиков данных, общее кол-во запросов по этим ключам, а также часть со списком действий агента с ключами.

Данный список будет содержать описание действия без явного указания ключа, например: «получен ключ», «ключ заблокирован», «ключ обновлён», дату и время действия и типы обработчиков данных, к которым относится ключ. У списка должна быть сортировка по дате и фильтры по типам обработчиков данных. В этом списке отображаются только непосредственные действия агента, последствия действий юнита с агентом не отображаются.

**4.5.1.3. Экран уведомления о блокировке агента**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_AGENT и на значение поля IS\_ACTIVATED, которое должно быть TRUE.

На этом экране сообщаем агенту, что он заблокирован, и при необходимости должен связаться со своим юнитом по e-mail (указываем e-mail связанного юнита).

**4.5.2. Вкладка управления данными юнита//+**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_UNIT и на значение поля IS\_ACTIVATED, которое должно быть TRUE.

На данном экране отображаются все данные, которые пользователь-юнит указывал при регистрации с отметками возле тех полей, которые возможно изменить. Некоторые поля (например, e-mail) изменить нельзя, для их изменения можно обратиться в техподдержку. Информацию об этом нужно разместить на текущем экране.

Если пользователь при регистрации активировал чекбокс “Free category account”, то здесь также отображается текущий статус проверки на принадлежность к категории бесплатных пользователей FREE\_STATUS: сразу после регистрации он будет PENDING, если будет подтвержден - поменяется на ACCEPTED, если отклонен - на DECLINED.

Также на этом экране будет отображаться общее кол-во привязанных пользователей-агентов. Значение будет активной ссылкой, которая перенаправляет на вкладку управления агентами.

На экране будет модуль с набором кнопок управления: «Изменить пароль» — перенаправляет на экран изменения пароля, «Изменить данные» - перенаправляет на экран изменения данных.

Также на вкладке присутствует отдельная область для уведомлений — она пуста при переходе после нажатия вкладки (нужна, чтобы не плодить попапы).

**4.5.2.1. Экран изменения пароля//+**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_UNIT и на значение поля IS\_ACTIVATED, которое должно быть TRUE.

На данном экране находится форма изменения пароля. Она состоит из поля ввода старого пароля, нового пароля и дублирующего поля нового пароля и кнопки «подтвердить изменение». Если валидация (происходит на стороне сервера) прошла успешно, то пользователь перенаправляется на вкладку управления данными юнита, в области уведомлений появляется сообщение «пароль успешно изменен». На e-mail приходит письмо с информацией о том, что пароль был изменен (без явного указания паролей!). В БД записываем дату и время последнего изменения пароля.

Если валидация не пройдена, то пользователь перенаправляется обратно на экран с формой, и возле проблемных полей показываем сообщения об ошибках.

**4.5.2.2. Экран изменения данных//+**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_UNIT и на значение поля IS\_ACTIVATED, которое должно быть TRUE.

На данном экране находится форма изменения данных. Она состоит из полей данных учетной записи (которые возможно изменить) и кнопки «подтвердить изменения». Если валидация (происходит на стороне сервера) прошла успешно, то пользователь перенаправляется на вкладку управления данными юнита, в области уведомлений появляется сообщение «данные успешно изменены». На e-mail приходит письмо с информацией о том, что данные были изменены. В БД записываем дату и время последнего изменения данных.

Если валидация не пройдена, то пользователь перенаправляется обратно на экран с формой, и возле проблемных полей показываем сообщения об ошибках.

**4.5.3. Вкладка управления платежами**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_UNIT и на значение поля IS\_ACTIVATED, которое должно быть TRUE.

Эта вкладка будет содержать информацию о количестве средств, которое будет списано с карты пользователя в конце расчетного периода (количество рассчитывается на основе выбранных пакетов услуг и текущей статистике по запросам с использованием всех ключей всех пользователей-агентов, связанных с юнитом).

Эта вкладка также будет содержать ссылки на формы заполнения финансовых данных клиента (формы будут предоставлены внешними сервисами обработки платежей), мы не будем их хранить у себя в БД. При успешном заполнении такой формы клиентом, мы получим от внешнего сервиса соответствующий токен, который позволит нам списывать средства.

Если платежная информация уже была заполнена и подтверждена внешним сервисом, то также должна быть кнопка “Отвязать карту” (для нее нужно создать модальное окно подтверждения “Вы уверены?”). После отвязки карты все активные ключи агентов этого юнита блокируются.

Также будет отображаться текущий статус возможности списания средств. Если платежная информация уже была заполнена и подтверждена внешним сервисом обработки платежей (т.е. у нас есть соответствующий токен для списания средств), то статус будет “Confirmed”, если информация не заполнена или не подтверждена, то статус будет “Not confirmed”, и рядом добавим пояснение, что нужно сделать, чтобы его изменить.

Также на вкладке будет кнопка «история платежей», которая перенаправляет на экран истории платежей.

**4.5.3.1. Экран истории платежей**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_UNIT и на значение поля IS\_ACTIVATED, которое должно быть TRUE.

Экран содержит список всех успешных платежей в системе с указанием даты и времени, суммы и системы оплаты. Также указаны ID для всех платежей, чтобы можно было сообщить его в случае связи с техподдержкой.

Также для каждого платежа есть возможность скачать файл ранее сгенерированного репорта по соответствующему периоду.

Для списка работает пагинация и сортировка по дате и времени, сумме платежа.

**4.5.4 – Вставить попАп «подключи карту»**

**4.6. Экран восстановления пароля //+**

Переход на этот экран не валидируется на наличие авторизации. Пользователь переходит на него с главного экрана сервиса из блока авторизации при нажатии на кнопку «забыл пароль».

Экран содержит описание процедуры восстановления и форму, которая состоит из поля ввода e-mail и кнопки «отправить». Валидация для этой формы происходит на стороне сервера, проверяется поле e-mail на соответствие строки стандартному паттерну e-mail.

Форма обязательно содержит поле с капчей!

Если валидация не пройдена, то пользователь перенаправляется обратно на экран с формой, и возле поля отображается сообщение об ошибке. Если валидация пройдена, то пользователь перенаправляется на экран уведомления о заявке на восстановление пароля, а на стороне бэка мы проверяем наличие такого e-mail в базе. Если такой e-mail найден, то отправляем на него письмо с сообщением о попытке восстановления пароля и персональную ссылку на восстановление пароля, она перенаправляет на экран формы восстановления пароля. Делаем запись даты и времени запроса в БД для соответствующего пользователя. Если e-mail не найден — не делаем ничего.

**4.6.1. Экран уведомления о заявке на восстановление пароля //+**

На этом экране мы показываем сообщение о том, что заявка на восстановление принята, и если такой e-mail зарегистрирован в системе, то на него будет отправлено письмо со ссылкой на восстановление, и что, в случае возникновения проблем, можно обратиться в техподдержку. Также будет кнопка «ОК», которая перенаправляет обратно на главный экран сервиса.

**4.6.2. Экран формы восстановления пароля//+**

Переход на этот экран не валидируется на наличие авторизации. Пользователь переходит на него по персональной ссылке из e-mail.

Экран содержит описание процедуры и форму, которая состоит из поля ввода нового пароля, дублирующего поля ввода пароля и кнопки «подтвердить». Валидация для этой формы происходит на стороне сервера, проверяется поле пароля на соответствие дублирующему полю и на длину строки.

Если валидация не пройдена, то пользователь перенаправляется обратно на экран с формой, где возле соответствующего поля отображается сообщение об ошибке. Если валидация пройдена, то меняем пароль в БД, авторизуем пользователя, и пользователь перенаправляется (в зависимости от роли) или на вкладку управления данными юнита, или на вкладку управления данными агента, где в области уведомления отображаем сообщение «пароль успешно изменен». Делаем запись даты и времени изменения пароля в БД для соответствующего пользователя.

**4.6. Экран активации аккаунта пользователя-агента //+**

Переход на этот экран не валидируется на наличие авторизации. Пользователь переходит на него по персональной ссылке из e-mail.

На этом экране кратко описана дальнейшая процедура активации. Также экран содержит форму, которая состоит из поля ввода нового пароля пользователя (свой пароль агент устанавливает сам, у связанного с ним пользователя-юнита нет доступа к его паролю), дублирующее поле пароля и кнопку «подтвердить».

Как и для формы регистрации — здесь должно быть обязательное поле-чекбокс для согласия с legal terms и ссылкой на них.

Валидация для этой формы происходит на стороне сервера.

Если валидация не пройдена, то пользователь перенаправляется обратно на экран с формой, и возле соответствующего поля отображается сообщение об ошибке. Если валидация пройдена, то пользователь перенаправляется на экран сообщения об успешной активации пользователя-агента (этот экран по сути полностью соответствует экрану сообщения об успешной активации пользователя-юнита, только текст будет описывать возможности агента). Делаем запись даты и времени активации в БД для соответствующего пользователя, и поле IS\_ACTIVATED для данного пользователя-агента становится TRUE.

**4.7. Главный экран области пользователя-агента //+**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_AGENT и на значение поля IS\_ACTIVATED, которое должно быть TRUE, или на наличие авторизации, роли ROLE\_UNIT, на значение поля IS\_ACTIVATED, которое должно быть TRUE, и на наличие привязки пользователя-агента, для которого строится экран, к текущему авторизованному юниту.

К этому экрану имеет доступ только авторизованный пользователь-агент (или пользователь-юнит, если экран построен для дефолтного агента). На этот экран можно перейти после активации аккаунта пользователя-агента или по кнопке «личный кабинет агента» (она будет доступна только авторизованным агентам) с главного экрана сервиса. В случае дефолтного агента - также с помощью перехода по вкладке “Управление ключами” на главном экране области привязанного к нему пользователя-юнита.

На этом экране доступны две вкладки: вкладка управления ключами (вкладка активна по умолчанию) и вкладка управления данными агента. Если этот экран построен для дефолтного пользователя-агента, то этот набор вкладок заменяется на набор вкладок с главного экрана области пользователя-юнита, к которому привязан дефолтный агент:

управление агентами, управление данными юнита, управление платежами, управление ключами (вкладка активна по умолчанию).

**4.7.1. Вкладка управления ключами**

На этой вкладке будет находиться список привязанных к данному агенту ключей API. Обязательно присутствует пагинация. Также есть сортировка по ID ключа, имени ключа, дате выдачи ключа, общему количеству отправленных запросов по этому ключу. Список можно фильтровать по ID ключа, имени ключа и по статусу ключа («активен», «блокирован»).

В списке присутствуют все вышеперечисленные поля, при этом имя ключа является активной ссылкой. При нажатии на нее, пользователь перенаправляется на экран управления ключом.

На экране есть область уведомлений, аналогичная по функционалу таким же областям на других экранах.

Также на экране есть форма добавления ключа, она состоит из чекбоксов выбора типов обработчиков данных (в списке только доступные для текущего агента), поля ввода имени ключа (в поле будет отображаться дефолтное значение, сформированное на основе количества созданных ключей для этого агента, которое агент сможет редактировать, имя ключа должно быть уникальным для данного агента) и кнопки «Добавить ключ». При нажатии на нее, с помощью модуля связи с API к пользователю-агенту привязывается новый ключ. Экран перезагружается, и новый ключ доступен в списке. В области уведомлений появляется сообщение «ключ успешно создан». Если платежные данные привязанного к агенту пользователя-юнита не были заполнены и подтверждены (или юнит был заблокирован за неуплату), то форма добавления ключа недоступна, кроме случая, когда привязанный юнит является бесплатным пользователем (см. описание раздела модуля работы с платежами).

Если экран построен для дефолтного агента, то сразу после регистрации привязанного юнита в списке ключей будет доступен демо-ключ (подробное описание работы демо-ключа в разделе модуля работы с платежами).

**4.7.1.1. Экран управления ключом**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_AGENT и на значение поля IS\_ACTIVATED, которое должно быть TRUE, или на наличие авторизации, роли ROLE\_UNIT, на значение поля IS\_ACTIVATED, которое должно быть TRUE, и на наличие привязки пользователя-агента, для которого строится экран, к текущему авторизованному юниту. Также обязательна проверка на наличие привязки этого ключа к агенту, по данным которого строится экран.

На этом экране будет модуль основной информации о ключе, он будет содержать строку с именем ключа, токеном ключа, дату и время создания ключа, текущий статус ключа, типы обработчиков данных, к которым относится ключ.

Также здесь будет кнопка “Загрузить файлы ключа”, при ее нажатии в систему пользователя будут загружены все файлы, относящиеся к текущему ключу.

Возле поля имени ключа будет кнопка “Переименовать”, при нажатии на нее отображаемое значение имени меняется на текстовое поле формы, содержащее текущее значение имени ключа, а кнопка “Переименовать” меняется на “Сохранить”. После редактирования и нажатия на кнопку “Сохранить” - сохраняем изменения в БД и вместо текстового поля снова отображаем строку с новым значением имени ключа.

Также здесь будет модуль управления ключом, он будет содержать несколько кнопок: кнопку блокировки ключа («блокировать»), при ее нажатии (применяем AJAX) — данный ключ агента блокируется (средствами модуля связи с API), статус блокировки ключа меняется на TRUE, а дальнейшие запросы по этому ключу не будут работать. В БД создается соответствующая запись.

Также в модуле будет кнопка обновления ключа, при ее нажатии текущий ключ блокируется, агенту взамен выдается новый ключ (содержащий доступы к тем же самым обработчикам данных), и пользователь перенаправляется на экран управления новым ключом. В БД создается соответствующая запись.

Все вышеперечисленные кнопки внутри модуля управления ключом обязательно вызывают попап с подтверждением «вы уверены?». Для кнопки «Блокировать» нужно добавить сообщение, что данное действие необратимо.

Также на этом экране будет список (по дням) статистики количества запросов по ключу. У списка должна быть сортировка по дате и фильтр по периоду времени (с точностью до дня).

Также отображаем соответствующий график (который перестраивается при изменении фильтра периода времени).

**4.7.2. Вкладка управления данными агента //+**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_AGENT и на значение поля IS\_ACTIVATED, которое должно быть TRUE.

На данном экране отображается e-mail агента. Поле e-mail изменить нельзя, для его изменения можно обратиться в техподдержку. Информацию об этом нужно разместить на текущем экране.

Также на этом экране будет отображаться e-mail пользователя-юнита, к которому привязан агент.

На экране будет кнопка «изменить пароль», она перенаправляет на экран изменения пароля. Функционал изменения пароля абсолютно соответствует этому процессу для пользователя-юнита.

Также на вкладке присутствует отдельная область для уведомлений — она пуста при переходе после нажатия вкладки.

**5. Раздел администратора (отсутствует в MVP)**

**5.4.2. Экран уведомления о блокировке юнита**

Переход на этот экран обязательно валидируется на наличие авторизации, роли ROLE\_UNIT и на значение поля IS\_ACTIVATED, которое должно быть TRUE.

На этом экране сообщаем юниту, что он заблокирован, и при необходимости должен связаться с техподдержкой.

**6. Раздел техподдержки**

В футере указываем e-mail для приема обращений в техподдержку.

**7. Модуль связи с API**

Этот модуль полностью относится к бэкэнду, здесь не будет описаний экранов.

Соединение с сервером API логики производится с помощью фреймворка gRPC (<https://grpc.io/docs/tutorials/basic/php.html>). Шифрование соединения с сервером и контроль доступа к нему осуществляется с помощью SSL-ключей. Доступ к обработчикам данных и к сервису создания ключей и аутентификации будет регулироваться с помощью токенов OAuth2.0.

Для реализации функционала с сервером логики API необходимо на стороне бэкэнда обеспечить вызов специальных методов (с передачей им необходимых данных) и обработку результатов ответов от логики API. Подробная спецификация методов и их параметров будет представлена в отдельном файле, здесь будет описана принципиальная схема.

Все обработчики данных, функционал выдачи ключей и их проверки на стороне логики API оформлены в виде отдельных сервисов. У каждого сервиса будет свой уникальный идентификатор. При отправке запроса на генерацию ключа (токена и файлов для ssl) мы указываем, к какому пользователю будет привязан ключ, а также набор идентификаторов сервисов, к которым мы предоставляем пользователю доступ.

Необходимо будет создать специального пользователя, к которому будет привязан ключ с доступом к сервису авторизации (AuthService), именно от лица этого пользователя мы будем отправлять запросы к этому сервису на генерацию новых ключей и блокировку старых. Никакой другой пользователь не сможет получить ключ с доступом к сервису авторизации.

Чтобы выдать ключ пользователю для доступа к обработчикам данных, нужно будет от лица специализированного пользователя (с доступом к сервису авторизации) отправить запрос в API логику: GenerateTokenRequest, в теле запроса будет информация о пользователе, которому мы открываем доступ, а также набор идентификаторов сервисов, к которым мы открываем доступ.

Как только мы получаем ответ метода GenerateTokenResponse, мы сохраняем все данные о ключе (токен и файлы) и создаем в БД соответствующую запись о новом ключе с привязкой к соответствующему пользователю.

По сути, взаимодействие бэкэнда с логикой API будет ограничиваться только генерацией и блокировкой ключей. Методы проверки ключей и вызова обработчиков данных использоваться не будут. Основные данные о ключах (дата и время создания, содержимое ключа, период до деактивации, принадлежность к пользователю, набор идентификаторов обработчиков, к которым есть доступ и т.д.) будут храниться локально в нашей БД сервиса, а всю информацию об использовании ключей нужно будет извлекать из БД сервера с логикой API.

**8. Модуль связи с базой данных API**

Этот модуль полностью относится к бэкэнду, здесь не будет описаний экранов.

На стороне бэкэнда нужно будет создать специальный класс, который будет отвечать за создание и прерывание соединения с БД на сервере логики API, а также будет содержать набор специализированных (для всех экранов, где будет использоваться эта информация) методов с запросами к БД и обработкой их результатов.

Запросы в БД на сервер логики API будут отправляться каждый раз, когда будут вызываться соответствующие экраны с данными.

Для сохранения статистики по дням, необходимо создать cron-команду, которая будет раз в сутки забирать актуальную статистику из БД сервера логики API и сохранять в локальной БД сервиса в соответствующих таблицах.

На случай, если БД на сервере логики API будет недоступна, нужно предусмотреть специальный механизм уведомления пользователей о проблеме на тех экранах, где эта информация будет отображаться. Также необходимо предусмотреть логирование ошибок соединения и ошибок запросов к БД API именно на стороне сервера нашего сервиса.

Параметры доступа к БД на сервере логики API, а также подробная структура таблиц будут описаны в отдельном файле.

**9. Модуль для работы с платежами**

Этот модуль полностью относится к бэкэнду, здесь не будет описаний экранов.

Мы будем работать с внешними сервисами обработки платежей (в дальнейшем - эквайрами). На данный момент - это Stripe и PayPal.

Как только пользователь-юнит регистрируется, у него есть определенный запас бесплатных действий: определенное количество обработок фреймов изображений и секунд аудио. Для реализации этого механизма, при регистрации нового пользователя-юнита его дефолтный агент автоматически получает специальный ключ API с возможностью доступа ко всем обработчикам данных. На стороне логики API, как только счетчик бесплатных действий достигнет лимита, ключ блокируется. Добавление нового ключа для дефолтного агента и приглашение новых агентов будет невозможным, пока пользователь не заполнит и подтвердит свои платежные данные в форме эквайра.

Как только платежные данные пользователя будут подтверждены, мы от эквайра получаем специальный токен, который сохраняем у себя в базе. Это означает, что теперь пользователь может запрашивать новые ключи дефолтным агентом или приглашать новых агентов.

Действия (запросы) по всем ключам подсчитываются на стороне логики API, и каждый месяц 1-го числа (в данный момент расчетный период времени - 1 месяц) мы со стороны бэкэнда обращаемся в базу данных на сервере логики API и получаем необходимую статистику за прошедший период времени. Затем, на основе данных этой статистики, подсчитываем количество средств для каждого пользователя, которое мы собираемся списать с его карты. Результаты расчетов мы сохраняем у себя в БД, а всем пользователям отправляем PDF репорты с подробным описанием их действий и стоимости за прошедший платежный период. Эти репорты мы также сохраняем у себя на сервере, чтобы к ним можно было получить доступ позже.

У пользователя после получения репорта есть 24 часа, чтобы обратиться в техподдержку или к администрации сервиса в случае проблем, все проблемы решаются в ручном режиме. Если все штатно, то по истечении 24 часов все необходимые суммы списываются (в ручном режиме с помощью специальной команды) с соответствующих пользователей (с помощью методов обращения к эквайру с использованием соответствующих токенов пользователей). Как только средства успешно списаны - отправляем пользователям соответствующие письма (с чеками, например).

У каждого пользователя-юнита в БД есть поле FREE\_STATUS. Списание средств и отправка репортов осуществляется только для пользователей, у которых это поле со значением DECLINED или PENDING. Если поле со значением ACCEPTED - не списываем средства и не отправляем репорты.

Если средства не получается списать со счета, то блокируем все ключи, которые связаны с агентами соответствующего юнита.