

Документация по проекту

“01. АКИ. Агрегатор площадок и услуг креативных индустрий Москвы”

# Содержание

[Содержание 2](#_9zwmff4tkr49)

[**Термины и определения 4**](#_y9ioociftpti)

[**Общие сведения 5**](#_oabjh94mkaxq)

[Наименование системы 5](#_x3ithv84ybts)

[Перечень организаций, участвующих в разработке системы 5](#_7sbzqngajs3r)

[Информация о проекте 5](#_ajc8wxkn0yd7)

[Проблема 5](#_f2velgyyleyn)

[Цель 6](#_msavb0ribb7e)

[**Основные функции Платформы 7**](#_tcqt2jwovvss)

[**Основные технические решения 8**](#_p6px7pu9t8m0)

[Принципы проектирования и разработки 8](#_tn18yqdtcuk0)

[Технологии разработки и исполнения 8](#_urquyp6b6h5j)

[Архитектура Платформы 11](#_yqrtzy97mq88)

[Физическая модель данных 13](#_muskhf7fgr8q)

[Модель данных микросервиса по работе с организациями 13](#_j3cov4dnssng)

[Модель данных микросервиса по работе со справочниками 13](#_nxdto9tooqgo)

[Модель данных микросервиса по работе с файлами 14](#_ploh71cf8p58)

[Модель данных микросервиса по работе с площадками 14](#_mgqtwbucvuf9)

[Модель данных микросервиса по работе с бронированиями 15](#_jweiery9xay6)

[Модель данных микросервиса по работе с атрибутами бронирований 16](#_hlgefhtusw9z)

[Модель данных микросервиса отзывов пользователей о площадках 16](#_243r7lj5gzdp)

[Документирование API 16](#_f5vu1165igpv)

[Обеспечение масштабируемости, отказоустойчивости и высокой доступности сервисов Платформы 17](#_b8e6udon2cvt)

[Архитектура безопасности 18](#_8ayqmwde7fz3)

[Методы аутентификации пользователей 18](#_d45fq6lhallb)

[Методы разграничения прав доступа пользователей 18](#_fxhwgb2dcb8n)

[**Функциональное описание 19**](#_zhe1bxhl4zzu)

[Пользовательские сценарии 19](#_8j5pk4e6839z)

[Авторизация пользователя платформы (ТЗ п. 4.2.1, п. 4.2.2) 19](#_s6rg5w94gbpx)

[Регистрация пользователя платформы (ТЗ п. 4.2.1) 19](#_mdjmx8uijpdg)

[Регистрация администратора платформы (ТЗ п. 4.2.2) 20](#_pqknquymjbct)

[Редактирование пользователем реквизитов ЮЛ в Профиле (ТЗ п. 4.3) 21](#_m7blxrw3uuh2)

[Редактирование пользователем данных профиля ФЛ (ТЗ п. 4.3) 21](#_jxthpkskussf)

[Подписка на новости 22](#_w8re2puwmpu1)

[Просмотр информации о платформе 22](#_xfovepghedsd)

[Просмотр информации о контактах сервиса 23](#_4hhh36oxbroa)

[Просмотр новостей сервиса 24](#_6gcyn5bxz3u3)

[Арендодатель 24](#_4plpnaql4xms)

[Создание и редактирование площадки (ТЗ п.4.4) 24](#_ypj9wiqpyr50)

[Удаление площадки 25](#_gha793ydiihq)

[Просмотр бронирований площадок (ТЗ п.4.1) 25](#_dgcw6336g4rr)

[Подтверждение бронирования (ТЗ п.4.1) 26](#_rc8921azk3bi)

[Арендатор 27](#_pgl194u9ij00)

[Работа с каталогом площадок (ТЗ п.4.5) 27](#_5xopnz4lq1uk)

[Бронирование площадки (ТЗ п.4.7) 28](#_3bma128h4fzv)

[Просмотр бронирований площадок (ТЗ п.4.1) 28](#_pco8gvxzphyo)

[Создание отзыва об арендованной площадке (ТЗ п.4.1) 29](#_d6pwksbnpgn0)

[Административные 29](#_q9kfbivmmr71)

[Подтверждение регистрации ЮЛ 29](#_lcbsyhhvlct0)

[Подтверждение площадки (ТЗ п.4.1) 30](#_9uqp3ih8cypz)

[Удаление площадки (ТЗ п.4.1) 31](#_pq9hxhjdung)

[Просмотр профилей пользователей (ТЗ п.4.1) 31](#_jsjo60bruy83)

[Просмотр профилей пользователей (ТЗ п.4.1) 31](#_pa94ze9js8wm)

[Ролевая модель 32](#_bkeo7p5aq3ca)

[Справочники 32](#_vjvzagykbu3z)

[Виды креативных площадок 32](#_3duf0alwkb9s)

[Статус записи 33](#_6y9owa2qxd8e)

[Статус площадки, описание 33](#_mfcuo7yse7gi)

[**Инструкция по развертыванию Платформы 34**](#_jyc9myrsqiqb)

[Состав дистрибутива 34](#_uffitt5p7m1e)

[Подготовительные операции 35](#_cu03551m11rz)

[Сборка docker образов 35](#_4hxuwrxtxds1)

[Первый запуск 36](#_bltddemdvd0g)

[Запуск 36](#_7p1btgvukrbz)

[Остановка 36](#_7nxnueun5bn8)

[Перезапуск 36](#_1m9d7zob15r6)

[**Реквизиты доступа к демо стенду 37**](#_sz7mwt9ls3bo)

[**Развитие платформы 37**](#_2nreh2qhnyx7)

[Функциональные требования 37](#_v5wr00oyniyn)

[Приложение в целом 38](#_692pmjrdfhbv)

[Сервис IAM 38](#_wgepyhscl4ii)

[Микросервисы 38](#_6ncqqn50mam6)

[Ролевая модель 39](#_tfu7u4c2e3re)

[**Релизы 40**](#_jlehmgcb6sjd)

[Релиз v1.0.0 40](#_9xjy9b2kg0pw)

[Релиз v1.1.0 40](#_y7ef64rl67o5)

[Описание доработок 40](#_t6fseuxfh4ko)

[Архитектурные изменения 40](#_noq535dl9bm2)

[Шлюз отправки уведомлений 41](#_5aj9a9dx0w48)

[Telegram-бот 41](#_ijvf3wskvese)

[Кэширование ответов сервисов фронтенда 42](#_o36v5f669xw6)

[Функциональные изменения 43](#_o7gwp3dlqxmc)

[Реализация формы показа новостей платформы 43](#_gzee123nprxr)

[Включение второго фактора авторизации (OTP) для пользователей Платформы 43](#_ajeenmtsm7un)

[Функция отправки сообщений электронной почты 44](#_al6975p2q5a2)

## 

# Термины и определения

|  |  |
| --- | --- |
| **Сокращение/Термин** | **Определение** |
| ВМ | Виртуальная машина |
| ПО | Программное обеспечение |
| ЮЛ | Юридическое лицо |
| API | Интерфейс прикладного программирования (англ. Application programming interface) |
| ASGI | ASGI (англ. Asynchronous Server Gateway Interface) — клиент-серверный протокол взаимодействия веб-сервера и приложения |
| IAM | Управление идентификацией и контролем доступа (англ. Identity and Access Management) |
| JWT | JSON Web Token (JWT) — JSON объект, который определен в открытом стандарте RFC 7519 |
| OTP | One Time Password (одноразовый пароль) |
| S3 | Simple Storage Service (S3) — облачный сервис, позволяющий хранить файлы любого типа и объема с REST API. |

# 

# Общие сведения

## Наименование системы

Полное наименование системы: Платформа бронирования креативных площадок

Краткое наименование системы: Платформа

## Перечень организаций, участвующих в разработке системы

Заказчик: Агентство креативных индустрий (далее АКИ).

## Информация о проекте

На данный момент не существует единой платформы (веб-сервиса) для бронирования креативных площадок и услуг (креативные кластеры, звукозаписывающие студии, галереи, киноплощадки для проведения съемок и кинотеатры для проведения показов и фестивалей и проч.), поэтому приходится искать данные о них из различных источников, что осложняет процесс как для горожан и бизнес-сообщества, так и для арендодателей и креативных площадок.

Создание веб-сервиса АНО “Агентство креативных индустрий” по онлайн бронированию в одном месте всех креативных площадок Москвы позволит собрать всю креативную инфраструктуру по «системе единого окна» (креативные кластеры, креативные пространства от звукозаписывающих студий до галерей, киноплощадки для проведения съемок и кинотеатры для проведения показов и фестивалей и проч.).

Платформа (веб-сервис) позволит удовлетворить запросы как горожан для организации своего тематического креативного досуга (например, аренда звукозаписывающей студии для записи своего музыкального трека), а также для бизнес-сообщества для организации своего рабочего процесса (например, перекрытие улицы для съемки фильма и организация кино-каравана на парковочной территории на перекрытой улице).

Сейчас поиск и бронирования площадок занимает очень много времени и ресурсов. Создание единого онлайн сервиса бронирования могло бы решить данную проблему.

### Проблема

Описание проблемы в формате “User Story”:

Наши **бизнес-пользователи** - представители креативных индустрий хотят проводить мероприятия на креативных площадках (находить, бронировать и контролировать проведение).

Но не могут, потому что сейчас **отсутствует единая платформа** бронирования креативных площадок.

А существующие решения, как правило, **являются узкоспециализированными** (например, сервис бронирования кабинета психологу) или не умеют предоставлять сервис юр. лицам.

И поэтому приходится искать данные о площадках **из различных источников**, что **осложняет и замедляет процесс** взаимодействия как для горожан и бизнес-сообщества, так и для арендодателей креативных площадок.

### Цель

Описание цели в формате “User Story”:

Для **арендаторов, бизнес-владельцев площадок (арендодателей), администраторов** (сотрудников АНО “Агентство креативных индустрий”) необходимо реализовать решение: “**Платформу бронирования креативных площадок**” - агрегатор площадок и услуг креативных индустрий Москвы, которое будет выполнять функцию единого сервиса поиска, бронирования и размещения креативных площадок.

Кроме удобства работы для пользователей, архитектура решения фокусируется на:

1. **Безопасности работы пользователей и доступа** к сервисам (у решения выделен отдельным компонентом сервер IAM для обеспечения защищенности авторизованного доступа пользователей платформы, а также API Gateway для защиты доступа к API сервисам.
2. **Интеграции** с внешними партнерами посредством API (с возможностью авторизованного доступа, обеспечиваемого API Gateway - сервисом публикации API).
3. **Гибкости развития и масштабирования** Платформы в виду ее микросервисной архитектуры и выделенных компонентов, обеспечивающих защиту решения.
4. **Механизме атрибутов** (тэгов) - что позволяет реализовать расширенный поиск и комплексную аналитику в дальнейшем.

# Основные функции Платформы

В Платформе должны быть реализованы следующие основные функции:

* Идентификация, аутентификация и авторизация пользователей (как единая точка управления доступом пользователей к [платформ](https://docs.google.com/document/d/1tx6MSCRGlJ4MhvoNrjD3Ioz_vHqu4Gqa6Q_8Vu0HtuI/edit#heading=h.mdjmx8uijpdg)е).
* Регистрацию пользователей платформы.
* Управление профилем пользователя и организации пользователя.
* Управление креативными площадками Арендодателя.
* Подтверждение бронирований площадок Арендаторами.
* Работа с каталогом площадок.
* Бронирование площадок Арендаторами.
* Создание отзывов о площадках Арендаторами.
* Подтверждение регистрации ЮЛ Администраторами.
* Подтверждение площадок Администраторами.
* Просмотр профилей пользователей Администраторами.
* Управление пользователями Администраторами.

# Основные технические решения

## Принципы проектирования и разработки

Принципы проектирования:

* проектирование Платформы «сверху вниз» – модульное нисходящее пошаговое проектирование алгоритма и структур данных;
* принцип модульности и абстракции;
* использование шаблонов проектирования;
* проектирование узкоспециализированных (highly cohesive) и слабо связанных (loosely coupled) модулей;
* использование готовых компонентов программного обеспечения;
* использование ПО с открытым исходным кодом.

При создании Платформы применяются следующие общесистемные принципы:

* принцип развития, предусматривающий в ПО возможность его наращивания и совершенствования компонентов и связей между ними;
* принцип информационного единства, то есть во всех, подсистемах, средствах обеспечения и компонентах ПО используются единые термины, символы, условные обозначения и способы представления;
* принцип совместимости, состоящий в том, что язык программирования, исходный код и средства программного обеспечения согласованы, обеспечивают совместное функционирование всех компонентов Платформы и сохраняют открытой структуру системы в целом.

Прикладное программное обеспечение Платформы имеет унифицированный пользовательский интерфейс.

Предусмотрена возможность дальнейшего развития и модернизации Платформы: после сдачи Платформы в эксплуатацию Заказчик имеет возможность самостоятельно, либо с привлечением подрядных организаций, осуществлять развитие и модернизацию Платформы, в том числе, расширение ее функций по следующим направлениям:

* ­расширение функциональных возможностей Платформы за счет дополнительной разработки и/или внедрения новых компонентов.
* ­увеличение количества пользователей при соответствующем горизонтальном масштабировании вычислительных мощностей технического обеспечения Платформы.

## Технологии разработки и исполнения

Выбор системного ПО осуществлялся только среди открытого ПО.

* Для развертывания системы выбрана платформа для разработки, доставки и запуска контейнерных приложений Docker. В рамках хакатона деплой осуществляется при помощи Docker compose. Развертывание в контейнерах является стандартом де-факто для данного класса задач по разработке.
* Для балансировки нагрузки и снятия TLS выбран высокопроизводительный веб-сервер nginx (самый широко используемый веб-сервер в мире с долей в 33,8%).
* Для управления идентификацией и контролем доступа пользователей выбрано ПО Keycloak. Позволяет гибко управлять пользователями, их ролями, подключать внешних поставщиков идентификации, добавлять второй фактор аутентификации, настраивать темы страниц регистрации и авторизации пользователей.
* Для хранения реляционных данных выбрана СУБД Postgresql. Postgresql - объектно-реляционная СУБД с открытым исходным кодом, поддерживает многоверсионность (MVCC), обладает высокой производительностью, масштабируемость. Широко используется в веб-разработке. В реестре отечественного ПО существует коммерческий аналог СУБД Postgres Pro, полностью совместимый с выбранной открытой версией.
* Для хранения файлов используется ПО min.io (открытая реализация S3 API). S3 API является стандартом де факто для организации доступа к файлам как объектам через REST API. В случае необходимости ПО min.io может быть заменен на аналогичные решения, реализующие S3 API с минимальными изменениями кода.
* В качестве API шлюза выбрано ПО krakend: высокая производительность по сравнению с аналогичными решениями, расширяемость и открытость кода.
* В качестве языка разработки фронтенда выбран широко распространенный фреймворк React 18 (ключевые библиотеки material ui 5, redux tool kit query 1.9.3).
* В качестве языка разработки бэкенда выбран python: скорость разработки, простой и понятный синтаксис, легкость в изучении, большой выбор фреймворков и библиотек. Широкий выбор специалистов на рынке труда.
* В качестве инструмента контроля версий схем БД и управления изменениями выбран фреймворк yoyo-migrations.
* Для реализации API выбран фреймворк fastapi (с веб-сервером uvicorn): высокая производительность, асинхронность, скорость и простота разработки.
* Для взаимодействия с СУБД Postgresql выбраны SQLAlchemy и psycopg2.

Технологии среды исполнения и системное ПО

|  |  |  |
| --- | --- | --- |
| **Технология / ПО** | **Версия** | **Описание** |
| docker | 23.0.5 | Платформа для разработки, доставки и запуска контейнерных приложений. Решение с открытым исходным кодом.  Лицензия: Apache License 2.0 |
| keycloak | 18.0.2 | Keycloak - продукт с открытым кодом для реализации технологи SSO (Single Sign-On) с возможностью управления доступом, реализуется с учетом современных стандартов безопасности.  Целью этого инструмента является сделать создание безопасных приложений и сервисов с минимальным написанием кода для аутентификации и авторизации.  Лицензия: Apache License 2.0 |
| krakend | 2.3 | API шлюз с открытым исходным кодом. Используется как решение для обеспечения подхода Zero Trust в текущей микросервисной архитектуре Платформы  Реализация:   * единой точки входа для клиентов API; * сокрытие топологии микросервисов; * rate limiting; * мониторинг; * логирование; * авторизация.   Лицензия: Apache License 2.0 |
| nginx | 1.24.0 | Веб-сервер и балансировщик нагрузки. Решение с открытым исходным кодом.  Лицензия: открытая, http://nginx.org/LICENSE |
| min.io | 2023.5.18 | Реализация файлового хранилища, совместимая с S3 API. Решение с открытым исходным кодом.  Лицензия: GNU AGPLv3 |
| postgresql | 15.3 | База данных. Решение с открытым исходным кодом.  Лицензия: открытая, PostgreSQL License |

Технологии разработки

|  |  |  |
| --- | --- | --- |
| **Технология / ПО** | **Версия** | **Описание** |
| React | 18.2.0 | Библиотека для языка программирования JavaScript с открытым исходным кодом для разработки пользовательских интерфейсов. Она помогает быстро и легко реализовать реактивность — явление, когда в ответ на изменение одного элемента меняется все остальное |
| python | 3.10.11 | Высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика. |
| fastapi | 0.95.2 | Веб-фреймворк для создания API, написанный на Python. FastAPI активно использует декораторы, аннотации типов и интроспекцию кода, что позволяет уменьшить количество шаблонного кода в веб-приложении. |
| uvicorn | 0.22.0 | ASGI совместимая реализация веб-сервера для Python |
| pydantic | 1.10.7 | Библиотека, построенная на основе классов данных Python и добавляющая дополнительные функции для валидации и анализа данных. Pydantic проверяет данные и управляет настройками с помощью аннотаций типов Python. |
| psycopg2 | 2.9.6 | Адаптер базы данных PostgreSQL для языка программирования Python. Его основными особенностями являются полная реализация спецификации Python DB API 2.0 и потокобезопасность (несколько потоков могут совместно использовать одно и то же соединение).  Psycopg2 в основном реализован на C как оболочка libpq, что делает его одновременно эффективным и безопасным. |
| SQLAlchemy | 2.0.15 | Программная библиотека на языке Python для работы с реляционными СУБД с применением технологии ORM. Служит для синхронизации объектов Python и записей реляционной базы данных. |
| yoyo-migrations | 8.2.0 | Инструмент миграции схемы базы данных. Миграции базы данных могут быть записаны в виде файлов SQL или сценариев Python. |

## Архитектура Платформы

Платформа реализована с учетом практики микросервисного подхода к проектированию и состоит из следующих компонентов:

* микросервис по работе с пользователями;
* микросервис по работе с организациями;
* микросервис по работе со справочниками;
* микросервис по работе с файлами;
* микросервис по работе с площадками;
* микросервис по работе с бронированиями;
* микросервис оповещений;
* микросервис по работе с атрибутами бронирований;
* микросервис отзывов пользователей о площадках;
* входящий балансировщик нагрузки;
* API шлюз;
* компонент идентификации и авторизации (IAM);
* S3 совместимое объектное хранилище файлов.

Концептуальная архитектура Платформы приведена на рисунке ниже.

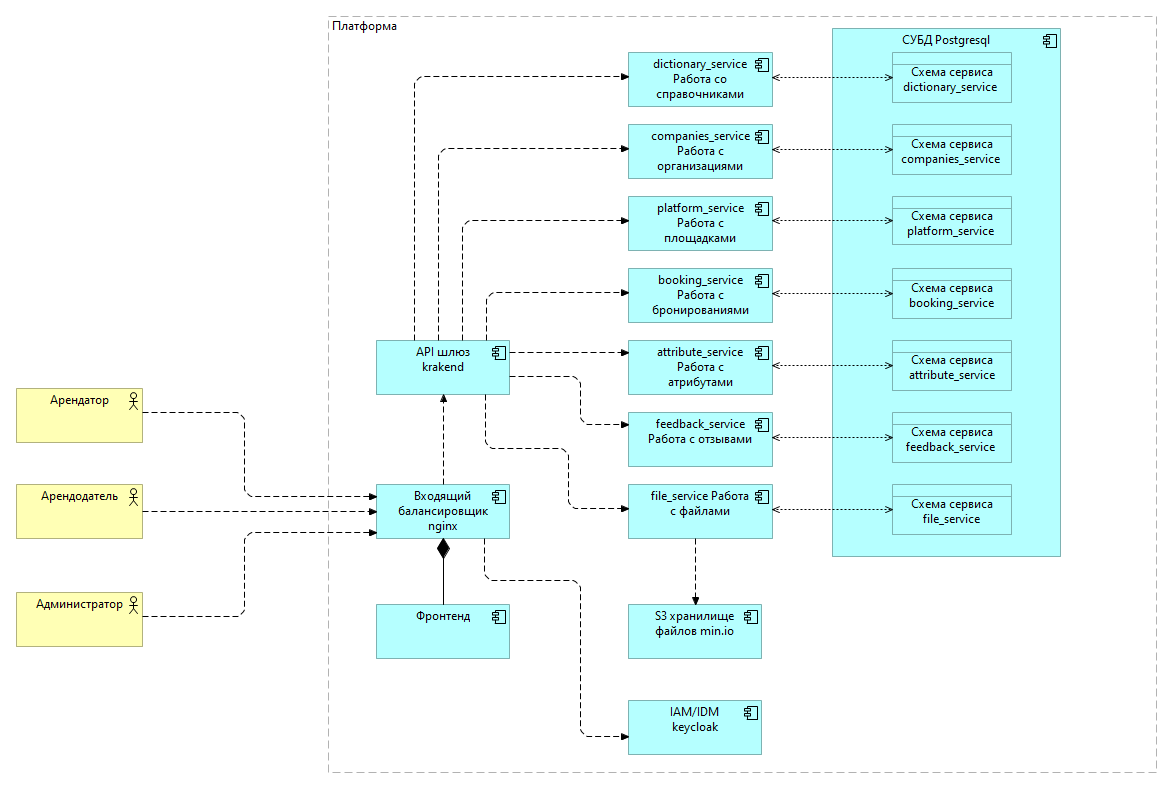


Рисунок 1. Концептуальная архитектура Платформы

Пользовательский трафик принимается на входящий балансировщик. Балансировщик снимает TLS и незашифрованный HTTP трафик перенаправляется на соответствующие компоненты. Фронтенд реализован по модели одностраничного приложения (SPA - Single Page Application). Бэкенд реализован в виде микросервисов, предоставляет REST API. Все сервисы бэкенда доступны только через API шлюз, который авторизует к ним доступ (совместно с IAM на базе Keycloak) и является единой точкой публикации методов сервисов. API шлюз также реализует rate-лимит обращений к методам сервисов.

Для хранения данных сервисов используется реляционная СУБД Postgresql. Каждый сервис хранит данные в выделенной схеме. Для хранения файлов используется ПО min.io (открытая реализация S3 API).

## Физическая модель данных

### Модель данных микросервиса по работе с организациями

Модель данных микросервиса представлена на рисунке ниже.

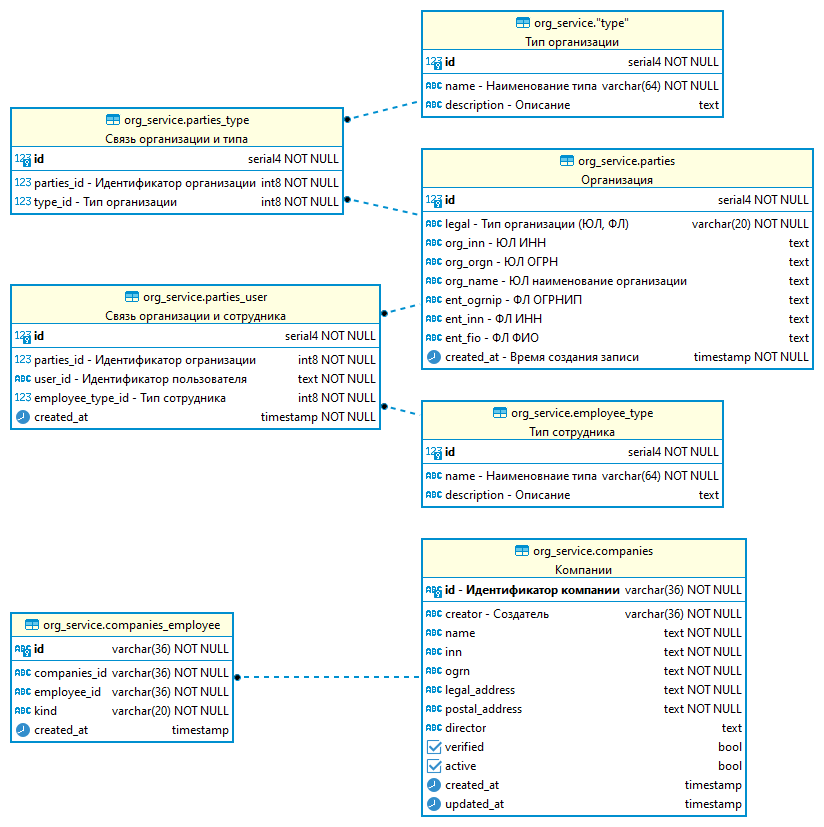


Рисунок 2. Модель данных микросервиса по работе с организациями

### Модель данных микросервиса по работе со справочниками

Модель данных микросервиса представлена на рисунке ниже.

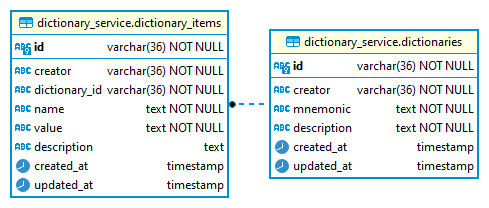


Рисунок 3. Модель данных микросервиса по работе со справочниками

### Модель данных микросервиса по работе с файлами

Модель данных микросервиса представлена на рисунке ниже.

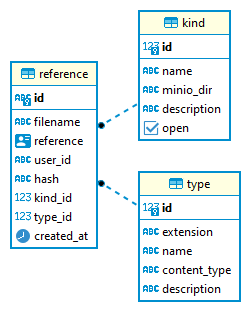


Рисунок 4. Модель данных микросервиса по работе с файлами

### Модель данных микросервиса по работе с площадками

Модель данных микросервиса представлена на рисунке ниже.

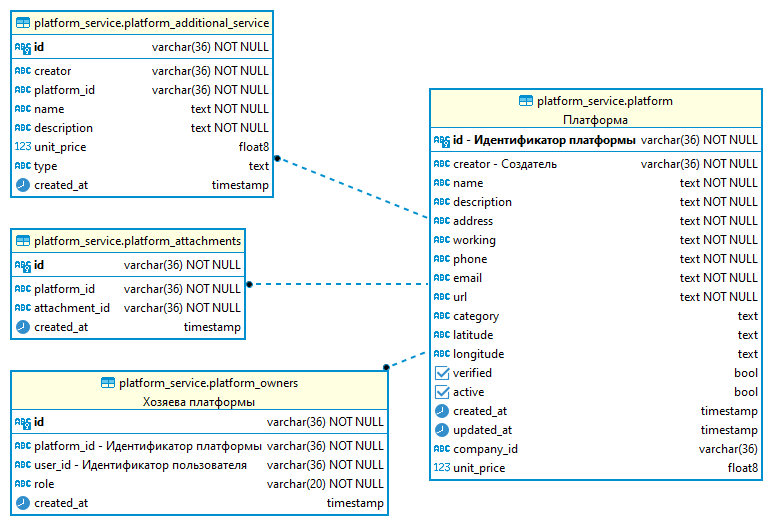
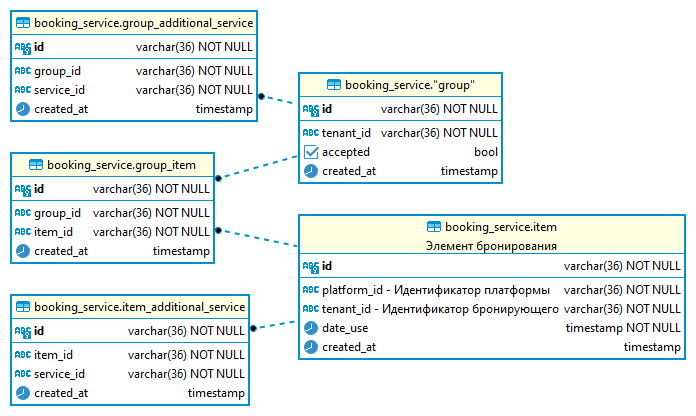


Рисунок 5. Модель данных микросервиса по работе с площадками

### Модель данных микросервиса по работе с бронированиями

Модель данных микросервиса представлена на рисунке ниже.

Рисунок 6. Модель данных микросервиса по работе с бронированиями

### Модель данных микросервиса по работе с атрибутами бронирований

Модель данных микросервиса представлена на рисунке ниже.

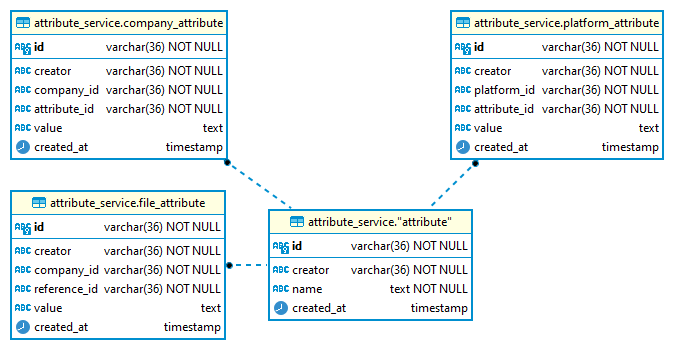


Рисунок 7. Модель данных микросервиса по работе с атрибутами бронирования

### Модель данных микросервиса отзывов пользователей о площадках

Модель данных микросервиса представлена на рисунке ниже.

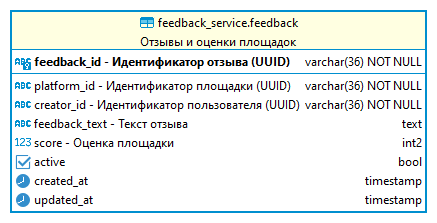


Рисунок 8. Модель данных микросервиса отзывов пользователей о площадках

## Документирование API

Каждый микросервис сопровождается документацией методов в формате OpenAPI Specification.

Документация для каждого сервиса доступна локально по URL [http://127.0.0.1:{port}/docs](about:blank)

Где {port} - соответствующий порт сервиса из файла конфигурации docker-compose.yml

В целях безопасности документация методов сервисов недоступна через API шлюз.

## Обеспечение масштабируемости, отказоустойчивости и высокой доступности сервисов Платформы

Целевой показатель нагрузки Платформы: до 100 тысяч пользователей.

В рамках хакатона развертывание решения осуществляется при помощи Docker compose: средства для определения конфигурации и запуска приложений Docker с несколькими контейнерами.

Для обеспечении масштабируемости, отказоустойчивости и высокой доступности сервисов Платформы при вводе в промышленную эксплуатацию рекомендуется развертывание в системе оркестрации контейнерова Kubernetes.

S3 хранилище файлов min.io позволяет реализовать отказоустойчивую кластерную конфигурацию, в т.ч. с возможностью геораспределенной репликации.

Масштабируемость, отказоустойчивость и высокая доступность СУБД Postgresql рекомендуется осуществлять путем добавления slave экземпляров СУБД, которые могут принимать на себя нагрузку на чтение и на них может быть осуществлено переключение пишущей нагрузки, в случае отказа master СУБД.

Масштабируемость, отказоустойчивость и высокая доступность IAM рекомендуется осуществлять путем запуска Keycloak в кластерном режиме. При необходимости система может быть развернута в катастрофоустойчивом режиме с балансировкой нагрузки и репликацией между ЦОД. IAM используется также для хранения пользователей, ролей и конфигураций БД Postressql. Размещение БД IAM рекомендуется выполнять вне кластера самого Keycloak. Для обеспечения отказоустойчивости Postressql использовать рекомендуется также кластер Patroni.

В рамках MVP используется один узел Keycloak с размещением на выделенной виртуальной машине.

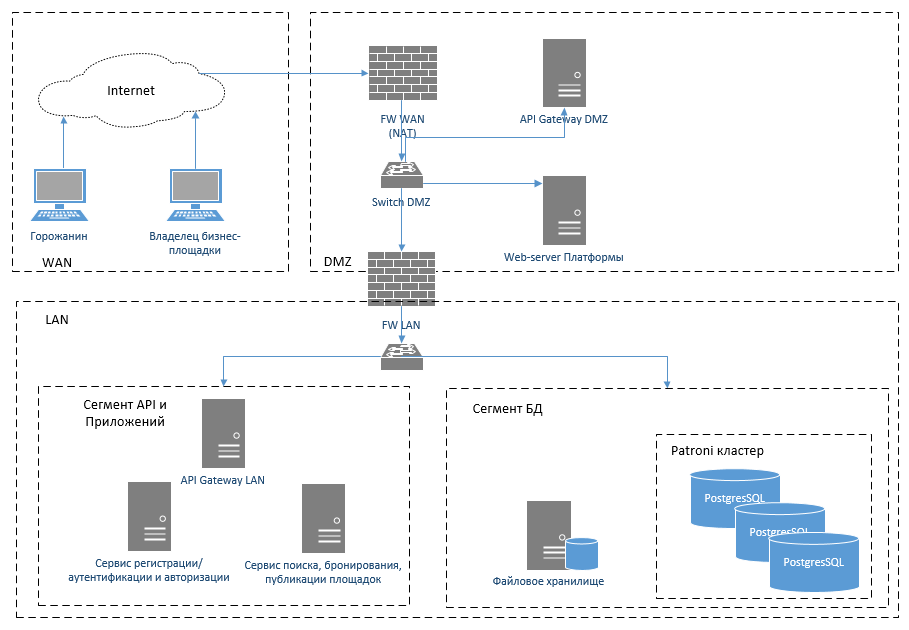


Рисунок 9. Рекомендуемая схема развертывания с обеспечением минимально достаточного уровня безопасности и отказоустойчивости

## Архитектура безопасности

### Методы аутентификации пользователей

Для аутентификации и авторизации пользователей в Платформе используются современные стандарты авторизации и аутентификации: OAuth 2.0, OpenID Connect (OIDC). Реализация: открытое ПО Keycloak.

В соответствии с рекомендациями фреймворка OAuth 2.0 для пользовательской авторизации при микросервисной архитектуре к использованию подлежит [Authorization Code Grant](https://tools.ietf.org/html/rfc6749#section-4.1) (c PKCE).

При аутентификации пользователя создается пользовательская сессия как на стороне приложения (основной подсистемы Платформы креативных площадок), так и на стороне сервера IAM с выдачей токена доступа (и токена обновления) в формате JWT.

Токен доступа имеет ограниченный период действия, по истечении которого потребуется запросить новый токен доступа (access token) посредством предоставления ранее полученного токена обновления (refresh token). Токен подписывается с использованием надежного алгоритма хэширования, что исключает подделку содержимого JWT токена при осуществлении проверки контрольной суммы на стороне API Gateway.

Проверка токена JWT аутентифицированного пользователя осуществляется в единой точке - API шлюзе. При проверке валидности JWT токена помимо целостности токена выполняется проверка системного clientId, зарегистрированного в API Gateway, чем обеспечивается подтверждение владельца токена.

Использование подхода с вынесением функций аутентификации и авторизации в отдельные сервисные компоненты обеспечения безопасности Платформы исключает возможность некорректной реализации проверок аутентификации разработчиками микросервисов и снижает дублирование программного кода.

А также обеспечивается этими же компонентами реализация подхода Zero Trust в текущей архитектуре Платформы.

### Методы разграничения прав доступа пользователей

Роль аутентифицированного пользователя добавляется в его токен доступа. Фронтенд и сервисы проверяют наличие соответствующих ролей для доступа к ресурсам.

Реализация выдачи ролей и проверки JWT токена: открытое ПО Keycloak и API Gateway.

Реализация ролевой матрицы доступа - Приложение (backend).

# Функциональное описание

## Пользовательские сценарии

### Авторизация пользователя платформы (ТЗ п. 4.2.1, п. 4.2.2)

**Актор**

Пользователь (любая роль).

**Предусловия**

* Открыта главная страница сервиса.
* Пользователь не авторизован.

**Основной сценарий**

* Пользователь нажимает кнопку “Войти” и переходит на форму авторизации.
* Пользователь вводит логин и пароль на форме.
* Пользователь нажимает кнопку “Войти”.

**Альтернативные сценарии**

* При последовательном переборе неправильных паролей, учетная запись блокируется на время (конфигурируемый параметр).

**Результат**

* Пользователь авторизован в платформе с соответствующей ролью.
* Пользователю доступна функциональность соответствующей роли.
* Открыта главная страница сервиса с функциональностью авторизованного пользователя.

### Регистрация пользователя платформы (ТЗ п. 4.2.1)

**Актор**

Пользователь

**Предусловия**

* Открыта главная страница сервиса.
* Пользователь не авторизован.

**Основной сценарий**

Пользователь нажимает кнопку “Регистрация”.

На форме регистрации пользователь заполняет поля:

* Фамилия
* Имя
* Отчество
* E-mail
* Номер телефона
* Пароль
* Подтверждение пароля

Пользователь выбирает роль в системе: Арендодатель или Арендатор.

Пользователь может ознакомиться с пользовательским соглашением и политикой конфиденциальности (при нажатии соответствующей ссылки документ открывается в отдельном окне).

Пользователь подтверждает свою электронную почту, указанную при регистрации (открывает указанную в письме ссылку).

Пользователь переходит на страницу заполнения реквизитов организации.

Пользователь заполняет реквизиты организации (см. сценарий редактирования реквизитов ЮЛ).

**Альтернативные сценарии**

Пользователь может вернуться на главную страницу сервиса.

**Результат**

* Пользователь зарегистрирован с указанной ролью.
* Эл. почта пользователя подтверждена.
* ЮЛ зарегистрировано и к нему привязан Пользователь.
* ЮЛ в статусе “Требует подтверждения”.

### Регистрация администратора платформы (ТЗ п. 4.2.2)

**Актор**

Суперадминистратор

**Предусловия**

Пользователь Администратор зарегистрирован.

**Основной сценарий**

Суперадминистратор

* открывает веб-интерфейс IAM.
* регистрирует пользователя с указанием эл. почты в качестве логина.
* отмечает эл. почту подтвержденной.
* устанавливает пользователю роль Администратора.

**Альтернативные сценарии**

**Результат**

* Пользователю назначена роль Администратора.

### Редактирование пользователем реквизитов ЮЛ в Профиле (ТЗ п. 4.3)

**Актор**

Пользователь

**Предусловия**

Пользователь зарегистрирован.

Открыта форма Редактирования реквизитов ЮЛ.

**Основной сценарий**

Пользователь заполняет реквизиты организации:

* Наименование
* ИНН
* ОГРН
* Почтовый адрес
* Юридический адрес
* Должность

Пользователь сохраняет изменения.

**Альтернативные сценарии**

Пользователь покидает форму, изменения не сохраняются.

**Результат**

* ЮЛ зарегистрировано и к нему привязан Пользователь.
* ЮЛ находится в статусе “Требует подтверждения”.
* Пользователь ЮЛ не может создавать/редактировать площадки.
* Пользователь ЮЛ не может бронировать.

### Редактирование пользователем данных профиля ФЛ (ТЗ п. 4.3)

**Актор**

Пользователь

**Предусловия**

Пользователь зарегистрирован, привязан к ЮЛ, эл. почта подтверждена.

Открыта форма Редактирования профиля ФЛ.

**Основной сценарий**

Пользователь заполняет данные профиля:

* Фамилия
* Имя
* Отчество
* E-mail
* Номер телефона
* Пароль
* Подтверждение пароля

Пользователь сохраняет изменения.

**Альтернативные сценарии**

Пользователь покидает форму, изменения не сохраняются.

**Результат**

* Данные профиля пользователя изменены.

### Подписка на новости

**Актор**

Пользователь

**Предусловия**

Открыта главная страница сервиса.

**Основной сценарий**

Пользователь:

* Выбирает тип интересуемой площадки.
* Заполняет E-mail.

Пользователь сохраняет изменения.

Пользователь подтверждает свою электронную почту, указанную при регистрации (открывает указанную в письме ссылку).

**Альтернативные сценарии**

Пользователь покидает форму, изменения не сохраняются.

Пользователь не подтверждает свою электронную почту, подписка не оформляется.

**Результат**

* Подписка на новости указанного типа площадок оформлена.

### Просмотр информации о платформе

**Актор**

Пользователь

**Предусловия**

* Открыта главная страница сервиса.

**Основной сценарий**

Пользователь нажимает кнопку “О платформе” и открывает форму.

Отображается форма с текстом.

**Результат**

Отображение контента:

<https://createdin.moscow/>

Агентство креативных индустрий является единым окном для взаимодействия

органов власти и представителей креативного предпринимательства Москвы.

Программы Агентства направлены на развитие и поддержку креативных

предпринимателей столицы.

Данный веб-сервис является единой платформой для

бронирования креативных площадок и услуг (креативные кластеры,

звукозаписывающие студии, галереи, киноплощадки для проведения съемок и

кинотеатры для проведения показов и фестивалей и проч.), что упростит процесс как для горожан и бизнес-сообщества, так и для арендодателей и креативных площадок.

Создание веб-сервиса АНО “Агентство креативных индустрий” по онлайн

бронированию в одном месте всех креативных площадок Москвы позволило собрать

всю креативную инфраструктуру по «системе единого окна» (креативные кластеры,

креативные пространства от звукозаписывающих студий до галерей, киноплощадки

для проведения съемок и кинотеатры для проведения показов и фестивалей и проч.).

Платформа позволяет удовлетворить запросы как горожан для

организации своего тематического креативного досуга (например, аренда

звукозаписывающей студии для записи своего музыкального трека), а также для

бизнес-сообщества для организации своего рабочего процесса (например,

перекрытие улицы для съемки фильма и организация кино-каравана на парковочной

территории на перекрытой улице).

### Просмотр информации о контактах сервиса

**Актор**

Пользователь

**Предусловия**

* Открыта главная страница сервиса.

**Основной сценарий**

Пользователь пролистывает страницу вниз, внизу страницы отображены контакты

**Результат**

Отображение контента:

<https://createdin.moscow/>

125009, Россия, Москва Вознесенский пер. 11. стр. 2

+7 495 988-30-80

info@moscow-creative.ru

### Просмотр новостей сервиса

**Актор**

Пользователь

**Предусловия**

* Открыта главная страница сервиса.

**Основной сценарий**

Пользователь нажимает кнопку “Все новости” и открывает форму.

Отображается форма с текстом.

**Результат**

Отображение контента (новости):

28.05.2023

Создание веб-сервиса АНО “Агентство креативных индустрий” по онлайн-бронированию в одном месте всех креативных площадок Москвы.

## Арендодатель

Юридические и физические лица, желающие разместить на платформе свою

креативную площадку.

### Создание и редактирование площадки (ТЗ п.4.4)

Размещать на платформе свои креативные площадки.

Обновлять информацию о площадке.

**Актор**

Пользователь

**Предусловия**

* Открыта форма просмотра списка площадок Арендодателя.
* Пользователь авторизован с ролью Арендодатель.

**Основной сценарий**

* Пользователь нажимает кнопку создать Площадку и переходит в карточку площадки
* Пользователь заполняет реквизиты Площадки:
  + Название
  + Описание
  + Адрес
  + Рабочие часы
  + Координаты
  + Категория
  + Дополнительные сервисы
  + Цена аренды
  + Телефон
  + Эл.почта
  + Ссылка на сайт
* Пользователь загружает файл презентации Площадки.
* Пользователь загружает файл оферты.
* Пользователь создает площадку.

**Альтернативные сценарии**

* Пользователь выбирает площадку на форме и переходит в карточку площадки.
* Пользователь редактирует реквизиты Площадки.
* Пользователь удаляет файлы Площадки, загружает новые файлы.
* Пользователь подтверждает изменения.

**Результат**

Площадка создана или данные площадки изменены. Площадка в статусе “Ожидает подтверждения”.

### Удаление площадки

**Актор**

Пользователь

**Предусловия**

* Открыта форма просмотра списка площадок Арендодателя.
* Пользователь авторизован с ролью Арендодатель.
* Отсутствуют активные бронирования.

**Основной сценарий**

* Пользователь нажимает кнопку удаления площадки.
* Пользователь подтверждает действие удаления в диалоговом окне.

**Альтернативные сценарии**

**Результат**

Площадка удалена (скрыта).

### Просмотр бронирований площадок (ТЗ п.4.1)

**Актор**

Пользователь

**Предусловия**

* Открыта форма просмотра просмотра списка площадок Арендодателя.
* Пользователь авторизован с ролью Арендодатель.

**Основной сценарий**

* Пользователь выбирает площадку (отображается название площадки, адрес, статус подтверждения) на форме и переходит в карточку активных бронирований.
* Пользователю отображается календарь с активными бронированиями.
* При нажатии на бронирование временного слота отображается форма информации об Арендаторе.

**Альтернативные сценарии**

* Для площадок со статусом “Отказано в подтверждении” пользователь может перейти на форму редактирования площадки, исправить данные площадки и заново отправить на подтверждение (см. сценарий Создание и редактирование площадки (ТЗ п.4.4)).

**Результат**

* Пользователь ознакомился с активными бронированиями и информацией об Арендаторах своей площадки.

**Ограничения реализации и развитие**

Выводить краткую статистику по загруженности площадок:

* кол-во занятых/свободных тайм-слотов за месяц/год.
* динамика изменения занятых/свободных тайм-слотов за месяц/год.
* результаты бронирований в разрезе площадок за выбранный отчетный период.

### Подтверждение бронирования (ТЗ п.4.1)

**Актор**

Пользователь

**Предусловия**

* Открыта форма просмотра активных бронирований.
* Пользователь авторизован с ролью Арендодатель.

**Основной сценарий**

* Пользователю отображается список запросов на бронирование
  + название площадки
  + временной слот
* Для каждого запроса бронирования Пользователь может ознакомиться с информацией об Арендаторе (соответствующая форма)
* Для каждого запроса бронирования Пользователь может перейти в расписание площадки
* Пользователь подтверждает бронирование

**Альтернативные сценарии**

* Пользователь отказывает в бронировании
* В запросе на бронирование проставляется статус “Отказано”

**Результат**

Бронирование подтверждено

## Арендатор

Юридические и физические лица, желающие забронировать представленные

на платформе площадки.

### Работа с каталогом площадок (ТЗ п.4.5)

**Актор**

Пользователь

**Предусловия**

* Открыта главная страница сервиса.
* Пользователь авторизован с ролью Арендатор

**Основной сценарий**

* Пользователю отображается карта Москвы с точками площадок
* Пользователь может настроить фильтр площадок:
  + Категория площадки
  + Временной интервал
  + Статус
* Пользователь выбирает площадку и смотрит краткую информацию о ней
  + Название
  + Адрес
  + Цена аренды
  + Название ЮЛ
* Пользователь нажимает кнопку “Подробнее” и переходит на форму бронирования площадки

**Альтернативные сценарии**

* Пользователь выбирает режим отображения списком.
* Пользователю отображается список площадок с краткой информацией
* Пользователь нажимает кнопку “Подробнее” и переходит на форму бронирования площадки

**Результат**

* Переход на форму бронирования

### Бронирование площадки (ТЗ п.4.7)

**Актор**

Пользователь

**Предусловия**

* Открыта форма бронирования площадки.
* Пользователь авторизован с ролью Арендатор.

**Основной сценарий**

* Пользователю отображается расписание площадки на ближайшие три месяца (если при работе с каталогом был указан фильтр по датам - данный диапазон полностью или частично должен попадать в отображаемые даты. дата начала фильтра попадает обязательно).
* Пользователь выбирает тайм-слоты для бронирования.
* Выбранные тайм-слоты выделяются цветом.
* При повторном нажатии на тайм-слот выбор с него снимается.
* Пользователь может переключить отображаемые месяцы вперед и назад по времени, но не ранее текущей даты.
* Даты в прошлом выделены цветом (серый) и недоступны для выбора.
* По завершении выбора пользователь переходит на форму подтверждения заказа бронирования (кнопка).
* Пользователь знакомится с договором офертой (ссылка на файл оферты площадки, открывается в отдельном окне).
* Пользователь отмечает чек-бокс “Принимаю положения договора оферты”.
* Пользователь отправляет бронирование на подтверждение Арендодателю (см. сценарий Подтверждение бронирования (ТЗ п.4.1)).

**Альтернативные сценарии**

**Результат**

* Бронирование в статусе “ожидает подтверждения” для выбранной площадки и диапазона дат создано.

**Ограничения реализации и развитие**

Развитие:

* тайм-слоты кратные получасу.
* учет часов работы.
* учет выходных дней.

### Просмотр бронирований площадок (ТЗ п.4.1)

**Актор**

Пользователь

**Предусловия**

* Открыта форма просмотра бронирований.
* Пользователь авторизован с ролью Арендатор.

**Основной сценарий**

* Пользователю отображается список его бронирований (завершенных, ожидающих подтверждения, подтвержденных).
  + Название площадки
  + Адрес площадки
  + Даты бронирования
  + Статус бронирования
  + Цена

**Альтернативные сценарии**

**Результат**

* Пользователь имеет возможность ознакомиться со своими бронированиями и статусом их подтверждения

### Создание отзыва об арендованной площадке (ТЗ п.4.1)

**Актор**

Пользователь

**Предусловия**

* Открыта форма просмотра бронирований.
* Пользователь авторизован с ролью Арендатор.

**Основной сценарий**

* Пользователю отображается список его бронирований (завершенных, ожидающих подтверждения, подтвержденных) (см. сценарий Просмотр бронирований площадок (ТЗ п.4.1)).
* Пользователь выбирает завершенное бронирование и нажимает кнопку создания отзыва.
* Открывается форма создания отзыва.
* Пользователь заполняет текст отзыва и ставит оценку площадке по пятибальной шкале.

**Альтернативные сценарии**

**Результат**

* Отзыв и оценка площадки сохраняются в сервисе.

## Административные

### Подтверждение регистрации ЮЛ

**Актор**

Администратор

**Предусловия**

* Пользователь авторизован с ролью Администратор.
* Открыта форма администрирования ЮЛ.

**Основной сценарий**

Администратору отображается список ЮЛ, требующих подтверждения регистрации

* Наименование
* ИНН

Администратор выбирает ЮЛ и система отображает ему форму информации о ЮЛ и ФЛ:

* Наименование
* ИНН
* ОГРН
* Почтовый адрес
* Юридический адрес
* ФИО
* Телефон
* Почта
* Должность
* Тип ЮЛ: Арендатор / Арендодатель

Администратор подтверждает регистрацию ЮЛ

**Результат**

* ЮЛ подтверждено.
* Пользователю, привязанному к ЮЛ доступны функции Арендатора / Арендодателя.

### Подтверждение площадки (ТЗ п.4.1)

**Актор**

Администратор

**Предусловия**

* Пользователь авторизован с ролью Администратор
* Открыта форма подтверждения площадки

**Основной сценарий**

* Администратору отображается список запросов на подтверждение площадки
  + название площадки
  + адрес площадки
* Для каждого запроса Администратор может ознакомиться с информацией об Арендодателе (соответствующая форма)
* Администратор подтверждает площадку

**Альтернативные сценарии**

* Администратор отказывает в подтверждении
* Для площадки проставляется статус “Отказано в подтверждении”

**Результат**

Площадка подтверждена.

### Удаление площадки (ТЗ п.4.1)

**Актор**

Администратор

**Предусловия**

* Пользователь авторизован с ролью Администратор.
* Открыта форма удаления площадки.

### Просмотр профилей пользователей (ТЗ п.4.1)

**Актор**

Администратор

**Предусловия**

* Пользователь авторизован с ролью Администратор.
* Открыта форма Просмотра профилей пользователей.

**Основной сценарий**

* Администратору отображается форма поиска и список пользователей
* В форме поиска Администратор может задать условие поиска по:
  + ФИО
  + Названию ЮЛ
  + ИНН
  + Эл.почте
* Список отображает краткую информацию о пользователях (с учетом фильтра):
* При нажатии на строку списка отображается форма полной информации (за исключением паролей) о пользователе и его ЮЛ (см. сценарии Регистрация пользователя платформы (ТЗ п. 4.2.1), Редактирование пользователем данных профиля ФЛ (ТЗ п. 4.3)).

**Результат**

* Администратор может найти полную информацию о пользователе сервиса.

### Просмотр профилей пользователей (ТЗ п.4.1)

**Актор**

Администратор

**Предусловия**

* Пользователь авторизован с ролью Администратор.
* Открыта форма Просмотра профилей пользователей.

**Основной сценарий**

* Администратору отображается форма поиска и список пользователей.
* В форме поиска Администратор может задать условие поиска по:
  + ФИО
  + Названию ЮЛ
  + ИНН
  + Эл.почте
* Список отображает краткую информацию о пользователях (с учетом фильтра):
* При нажатии на строку списка отображается форма полной информации (за исключением паролей) о пользователе и его ЮЛ (см. сценарии Регистрация пользователя платформы (ТЗ п. 4.2.1), Редактирование пользователем данных профиля ФЛ (ТЗ п. 4.3)).

**Результат**

* Администратор может найти полную информацию о пользователе сервиса

# Ролевая модель

В Платформе реализована следующая ролевая модель

|  |  |
| --- | --- |
| **Мнемоника роли в токене доступа** | **Роль** |
| ROLE\_TENANT | Арендатор |
| ROLE\_LANDLORD | Арендодатель |
| ROLE\_ADMIN | Администратор |

### Справочники

#### Виды креативных площадок

|  |  |
| --- | --- |
| **Мнемоника** | **Значение** |
| ART | Арт |
| ARCHITECTURE | Архитектура |
| GAMES\_SOFTWARE | Видеоигры и ПО |
| DESIGN | Дизайн |
| FILMS\_CREATING | Издательское дело и новые медиа |
| PERFORMING\_ARTS | Исполнительские искусства |
| FILM\_ANIMATION | Кино и анимация |
| FASHION | Мода |
| MUSIC | Музыка |
| ADVERTISEMENT | Реклама |

#### Статус записи

|  |  |
| --- | --- |
| **Мнемоника** | **Значение** |
| ACTIVE | Активна |
| BLOCKING | Заблокирована |
| UNCONFIRMED | Не подтверждена |
| CLOSED | Закрыта |

#### Статус площадки, описание

|  |  |
| --- | --- |
| **Мнемоника** | **Значение** |
| ANY | Любой |
| FREE | Свободна |
| BUSY | Занята |

#### 

# Инструкция по развертыванию Платформы

## Состав дистрибутива

Размещение: <https://github.com/BasePractice/htone-2023>

| **Путь** | **Описание** |
| --- | --- |
| conf/nginx | Файлы конфигурации входящего балансировщика nginx |
| src/backend | Sources root для сервисов бэкенда |
| src/backend/.gateway | Настройки API шлюза |
| src/backend/.requests | Тест-кейсы запросов к API по ключевым сценариям проверки методов |
| src/backend/api/model | Модели объектов API |
| src/backend/migrations | Скрипты миграции БД |
| src/backend/db | Модели объектов БД |
| src/backend/attribute\_service | Микросервис по работе с атрибутами бронирований |
| src/backend/booking\_service | Микросервис по работе с бронированиями |
| src/backend/companies\_service | Микросервис по работе с организациями |
| src/backend/dictionary\_service | Микросервис по работе со справочниками |
| src/backend/feedback\_service | Микросервис отзывов пользователей о площадках |
| src/backend/file\_service | Микросервис по работе с файлами |
| src/backend/platform\_service | Микросервис по работе с площадками |
| src/backend/user\_service | Микросервис по работе с пользователями |
| src/frontend | Sources root для фронтенда |
| src/frontend/build | Сборка фронтенда |
| https://www.keycloak.org/archive/downloads-18.0.2.html | IAM - открытый дистрибутив Keycloak. Для решения рекомендуется 18 версия. |
| https://github.com/BasePractice/htone-2023/blob/main/src/frontend/build/themes.zip | Стилизация и кастомизация Keycloak. |

## Подготовительные операции

Для развертывания MVP - Платформы необходима ВМ, удовлетворяющая следующим минимальным требованиям:

* vCPU: 2 шт
* vRAM: 4 Гб
* vHDD: 80 Гб
* ОС: Linux
* Доступ в интернет
* Публичный IP адрес и доменное имя (для доступа пользователей из Интернет)
* Открытый порт
  + HTTPS: 443 (для доступа пользователей из Интернет)
  + SSH: 22 (для настройки и сопровождения Платформы)
* Установленные пакеты:
  + docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
  + git
  + nginx
  + python3-certbot-nginx
  + zip unzip
  + curl

Загрузить исходные коды Системы на сервер и перейти в созданную папку дистрибутива

|  |
| --- |
| git clone <https://github.com/BasePractice/htone-2023.git>  cd ./htone-2023 |

Скопировать конфигурацию nginx из conf/nginx в целевую папку.

Выпустить сертификат TLS на требуемый домен, разместить в ВМ и прописать в конфигурации nginx

Распаковать сборку фронтенда из src/frontend/build в целевую папку nginx (в предоставленном дистрибутиве это /usr/share/nginx/ht2023)

Перезапустить nginx.

IAM разворачивается на выделенной vm в соответствии рекомендациями самого продукта <https://github.com/keycloak/keycloak-quickstarts/blob/latest/docs/getting-started.md>

Далее в папке themes добавляются стили с кастомизацией под Платформу.

## Сборка docker образов

Подключиться к серверу для развертывания Платформы

Перейти в папку дистрибутива, затем перейти в папку src/backend

Собрать docker образы, выполнив команду

|  |
| --- |
| ./dk\_build.sh |

## Первый запуск

Подключиться к серверу для развертывания Платформы

Перейти в папку дистрибутива, затем перейти в папку src/backend

В корне папки располагается файл конфигурации docker-compose.yml

Настроить параметры окружения в файле ./.env

Запустить Платформу

|  |
| --- |
| docker compose up |

Из файла src/backend/docker-compose.yml получить реквизиты подключения к СУБД (сервис postgres-db)

Подключиться к СУБД, создать БД

|  |
| --- |
| create database htone\_042023 |

Перезапустить Платформу

|  |
| --- |
| docker compose restart |

## Запуск

|  |
| --- |
| docker compose start |

## Остановка

|  |
| --- |
| docker compose stop |

## Перезапуск

|  |
| --- |
| docker compose restart |

# 

# Развитие платформы

В рамках технического задания нами был реализована функциональность, удовлетворяющая базовым требованиям заказчика. Понимая проблематику вопроса, для более полного, комплексного решения поставленных Агентству креативных индустрий задач, нами были проработаны предложения по развитию Системы, изложенные ниже.

### Функциональные требования

Реализовать следующие функциональные возможности:

* Возможность для Арендодателя указывать рабочие часы площадки и выходные дни (явно или по правилам, например, все официальные выходные в РФ), которые будут учитываться при бронировании.
* В зависимости от времени, дней недели и праздничных дней будет дана возможность назначения различной стоимости аренды.
* Расширение функций фильтрации и поиска.
* Будет реализована возможность интеграции с календарями Арендодателей.
* Генерация договоров с автоматической вставкой необходимых реквизитов и логотипа компании и последующим подписанием их с использованием ЭЦП.
* Дополнение карточки площадки возможностью предложения дополнительных услуг с использованием тематических иконок.
* Гибкая настройка календаря площадки - минимальные/максимальные периоды бронирования.
* Возможность расширения функциональности сервиса для более широкого круга лиц путём добавления новых видов креативных пространств.
* Возможность создания не только площадки для сдачи в аренду, но и мероприятия, в котором смогут участвовать как другие представители бизнес-сообщества предоставлением своих услуг, так и горожане для организации своего тематического креативного досуга.
* Предусмотреть возможность сдачи площадки не целиком, а части её (один из залов/цехов/кабинетов).

### Приложение в целом

Оптимизация исходного кода, CI/CD:

* Вынести все параметры настройки в соответствующие файлы конфигурации. Убрать из репозитория секреты. Сделать параметризацию конфигов для сред разработки, тестирования и прод.
* Провести рефакторинг логирования.
* Реализовать методы healthcheck или (liveness & readiness probes в случае деплоя в k8s).
* Оптимизировать сборку контейнеров (исключить повторную сборку, если зависимые библиотеки не меняются)

### Сервис IAM

Включение второго фактора авторизации для пользователей Платформы:

* OTP.
* СМС (интеграция с СМС-шлюзом).

Интеграция с внешними Identity провайдерами:

* mos.ru (получены первичные данные для настройки на тестовом стенде)
* ЕСИА.

Интеграция с Active Directory/Иным хранилищем учетных данных заказчика для авторизации Администраторов Платформы.

### Микросервисы

Реализовать следующие микросервисы для расширения функциональных возможностей.

Сервис статистики:

* статистика по загруженности площадок
* кол-во Арендаторов, Арендодателей
* активность (кол-во бронирований)

В разрезе Платформы в целом и для конкретных пользователей, временных интервалов.

Сервис оповещений

* Подтверждение бронирования Арендодателем.
* Новые запросы на Подтверждение бронирования.
* Новые запросы на Подтверждение ЮЛ.
* Новые запросы на Подтверждение площадок.

### Ролевая модель

Расширить ролевую модель следующими ролями:

* Администратор АКИ (все полномочия и назначает операторов АКИ).
* Оператор АКИ (проверки и подтверждения ЮЛ, Площадок, работа с обращениями пользователей).
* Администратор ЮЛ (все полномочия и назначает операторов Арендатора/арендодателя.
* Оператор АКИ (соответствующая функциональность Арендатора/арендодателя по работе с площадками и бронированиями).

# Релизы

## Релиз v1.0.0

Версия от 28.05.2023, отправленная на техническую экспертизу хакатона ЛЦТ 2023.

<https://github.com/BasePractice/htone-2023/tree/v1.0.0>

## Релиз v1.1.0

Версия от 09.06.2023, с доработками к финалу хакатона.

https://github.com/BasePractice/htone-2023/releases/tag/v1.1.0

### Описание доработок

#### Архитектурные изменения

Обновленная концептуальная архитектура Платформы представлена на рисунке ниже.

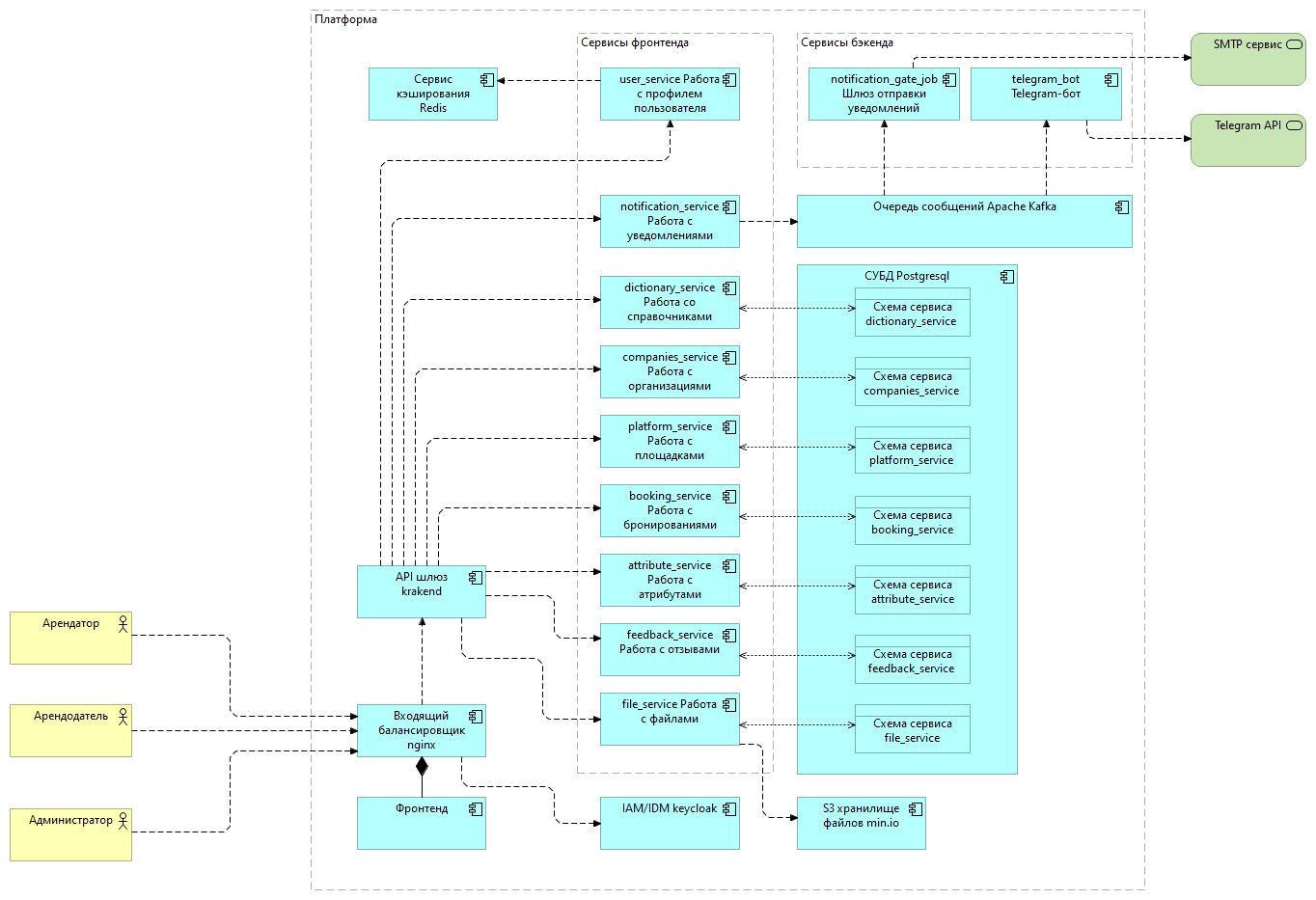


Рисунок 10. Концептуальная архитектура версии v1.1.0

Сервисы, которые вызываются со стороны SPA (фронтенд-приложение, исполняемое в браузере пользователя), выделены в концептуальную группу “Сервисы фронтенда”.

Введена новая группа “Сервисы бэкенда”. Данная группа сервисов не предполагает прямых вызовов из фронтенд-приложения. Предназначена для выполнения системных фоновых задач на основе запуска задач по расписанию или инициации запуска по системным событиям.

Обмен системными событиями реализован при помощи нового инфраструктурного компонента: очереди сообщений на базе Apache Kafka.

Apache Kafka — распределенный программный брокер сообщений с открытым исходным кодом, разрабатываемый в рамках фонда Apache на языках Java и Scala. Цель проекта — создание горизонтально масштабируемой платформы для обработки потоковых данных в реальном времени с высокой пропускной способностью и низкой задержкой

Введение очереди сообщений позволяет:

* развязать сервисы фронтенда и обработчики фоновых задач
* позволить отдельно масштабировать сервисы фронтенда и обработчики фоновых задач в зависимости от нагрузки на них
* обеспечить гарантированную доставку сообщений

В составе группы “Сервисы бэкенда” реализованы два сервиса:

* шлюз отправки уведомлений
* telegram-бот

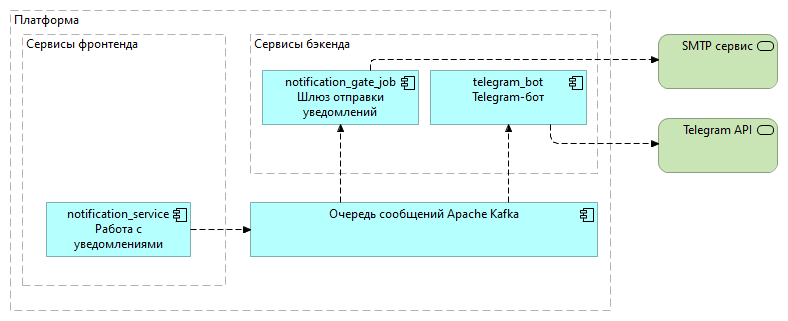


Рисунок 11. Реализованные сервисы бэкенда

##### Шлюз отправки уведомлений

Шлюз создан для реализации функции отправки уведомлений пользователям. В текущей реализации позволяет отправлять сообщения электронной почты (см. ниже описание функциональных изменений). Для этого реализована интеграция с сервисом SMTP.

В дальнейшем может быть реализована отправка сообщений по другим каналам доставки (СМС, пуш-уведомления).

Для инициации уведомлений из веб-приложения создат REST-сервис работы с уведомлениями. Сервис принимает запрос на отправку уведомления, проводит авторизацию, сохраняет уведомление в очередь сообщений.

##### Telegram-бот

Функциональность бота:

1. Регистрация пользователя: пользователь регистрируется в боте, идентификатором пользователя является электронная почта. Бот проверяет что в системе есть пользователь с такой же почтой (IAM) и отправляет проверочный код по почте, пользователь отдает код боту и он привязывает идентификатор пользователя телеграмм к пользователю Платформ.

2. Оповещение: бот посылает зарегистрированному пользователю оповещение о действиях в системе, связанных с пользователем (например, подтверждение компании-арендодателя или площадки). Настройка получения оповещения может быть реализована в профиле пользователя.

Примечание: функциональность реализована частично.

##### Кэширование ответов сервисов фронтенда

Согласно нефункциональным требованиям, целевой показатель нагрузки Платформы до 100 тысяч пользователей.

Для уменьшения нагрузки на сервисы, к которым часто требуется обращаться из веб-приложения или данные которых редко изменяются, целесообразно кэшировать их ответы на заданный интервал времени.

Для этой цели в Платформе был реализован механизм кэширования запросов к сервису работы с пользователями.

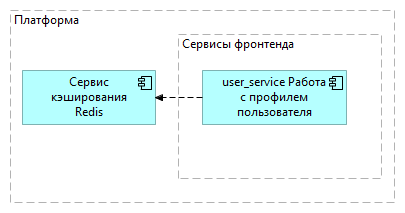


Рисунок 12. Кэширование

Данный сервис был выбран, поскольку запрос получения данных пользователя самый долгий (запрос идет в IAM для получения пользователей). Пользователь имеет уникальный идентификатор и поэтому мы не получим ложные срабатывания. Лаг по времени в 5 - 10 минут (время жизни записи кэша, настраиваемый параметр) приемлем при обновлении данных пользователя (роль, ФИО) и обновление этих данных происходит очень редко.

Кеширование происходит следующим образом:

* если это запрос по идентификатору, то он является ключом
* если идет запрос без идентификатора (список пользователей) мы его создает и однозначно сопоставляем с вызовом
* выполняем запрос в редис по ключу:
  + если ключ не найден (что может быть в двух случаях - никогда не обращались к кешу или TTL ключа истек) мы делаем запрос в IAM на получение данных. Затем сериализуем их в JSON и сохраняем в Redis по ключу (выставляя при этом TTL) и возвращаем результат.
  + если ключ найден - мы сразу же возвращаем результат

Проведен нагрузочный тест: время отклика метода получения пользовательских данных сократилось в 10 раз.

#### Функциональные изменения

##### Реализация формы показа новостей платформы

При переходе на форму новостей пользователю отображается список новостей платформы.

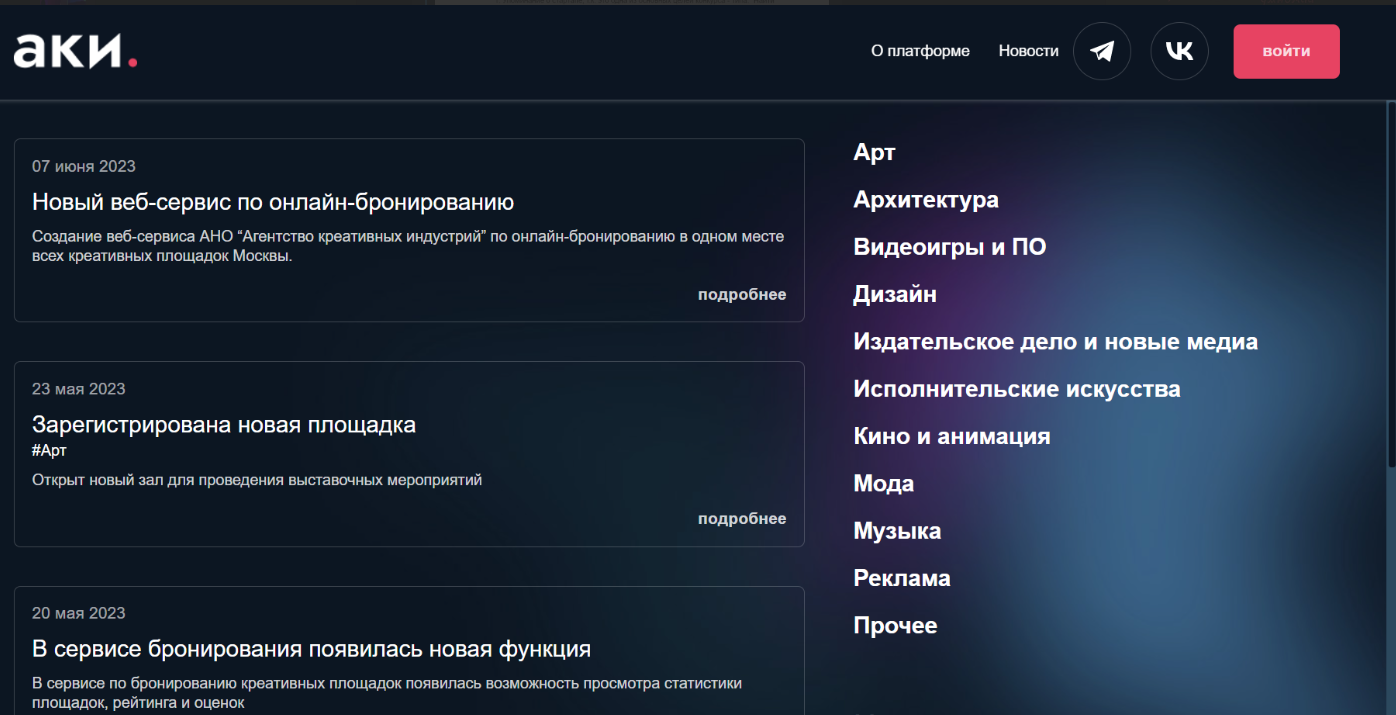


Рисунок 13. Форма новостей

Для каждой новости отображаются следующие поля:

* заголовок
* краткое описание
* описание
* категория

Реализована возможность фильтрации новостей по категории.

##### Включение второго фактора авторизации (OTP) для пользователей Платформы

Для авторизации в системе пользователю, кроме логина и пароля, необходимо ввести OTP.

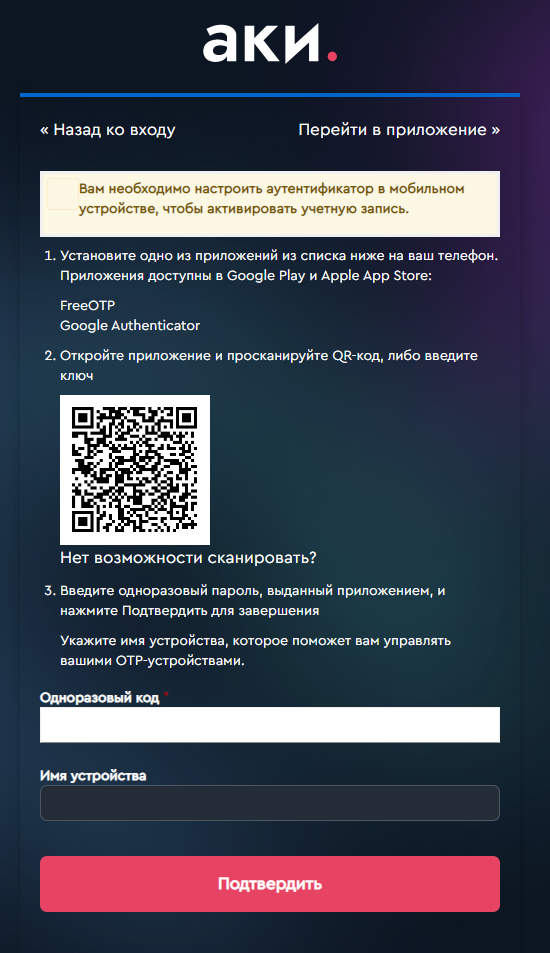


Рисунок 14. Настройка 2-го фактора (OTP - кода)

В данном релизе функция включена для администраторов АКИ по умолчанию. Также для пользователей возможность включения данной опции самостоятельно в личном кабинете присутствует.

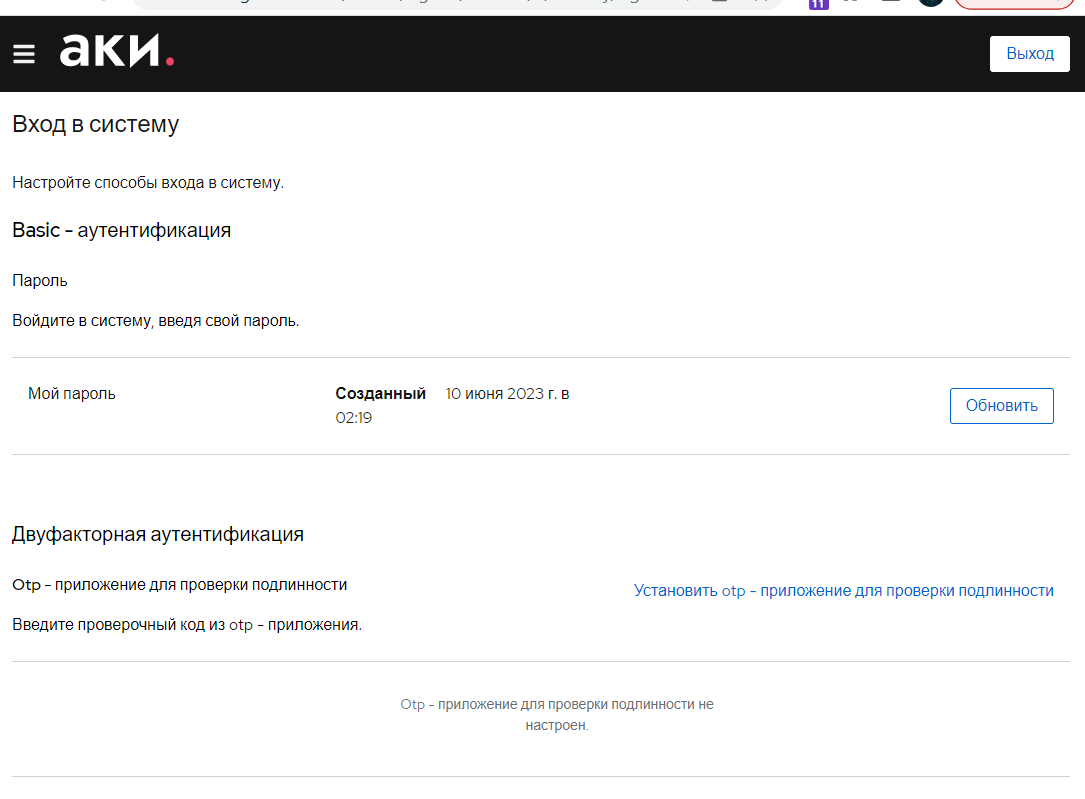


Рисунок 15. Настройка 2-го фактора (OTP - кода) из профиля пользователя

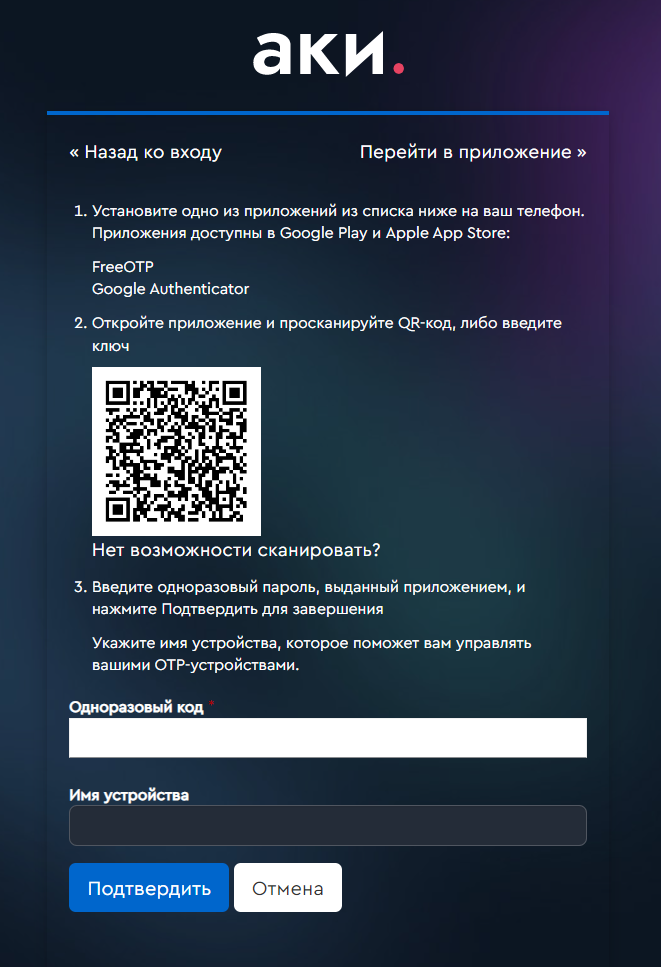


Рисунок 16. Настройка аутентификатора 2-го фактора (OTP - кода) из профиля пользователя

Для Администраторов системы OTP является обязательным в целях повышения безопасности платформы.

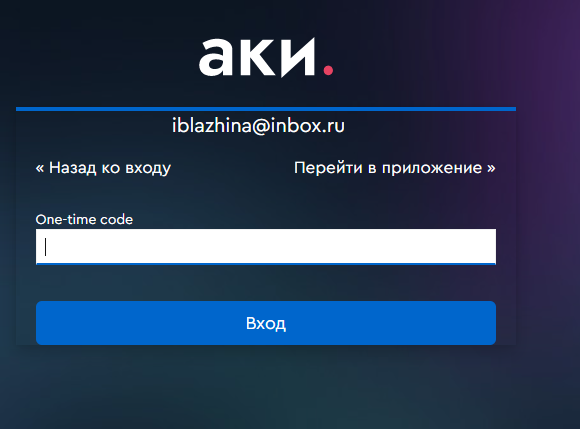


Рисунок 17. Подтверждение 2-го фактора (OTP)

#### Функция отправки сообщений электронной почты

Реализована возможность отправки сообщений электронной почты. Отправка сообщений может быть инициирована от авторизованного пользователя (только на свой e-mail), администратора (любому пользователю) и системным событием.

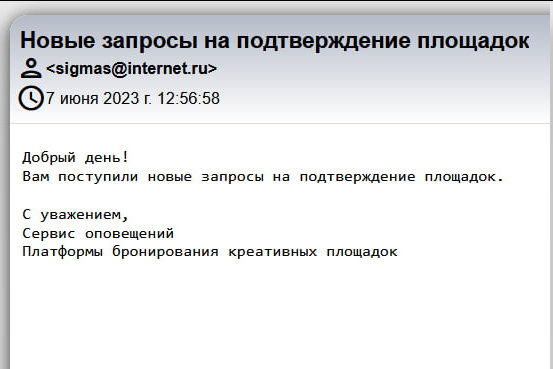
****

Рисунок 18. Оповещения по e-mail

Данная функциональность позволит реализовать оповещения пользователей для широкого круга событий:

* создание новых площадок
* появление новых запросов на подтверждение площадок, бронирований
* смена статусов бронирований (например, подтверждено Арендодателем)
* событий безопасности (вход из нового браузера/местоположения)

#### Функция аутентификации посредством mos.ru

Проведена настройка интеграции с системой СУДИР (mos.ru) для возможности ее подключения как внешнего провайдера. Для тестового стенда.

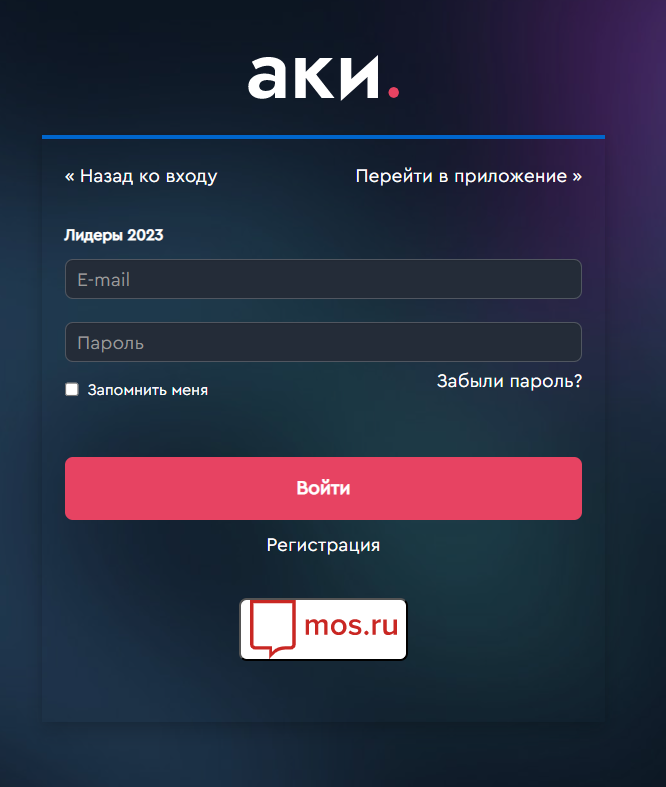


Рисунок 19. Добавление провайдера аутентификации **СУДИР**

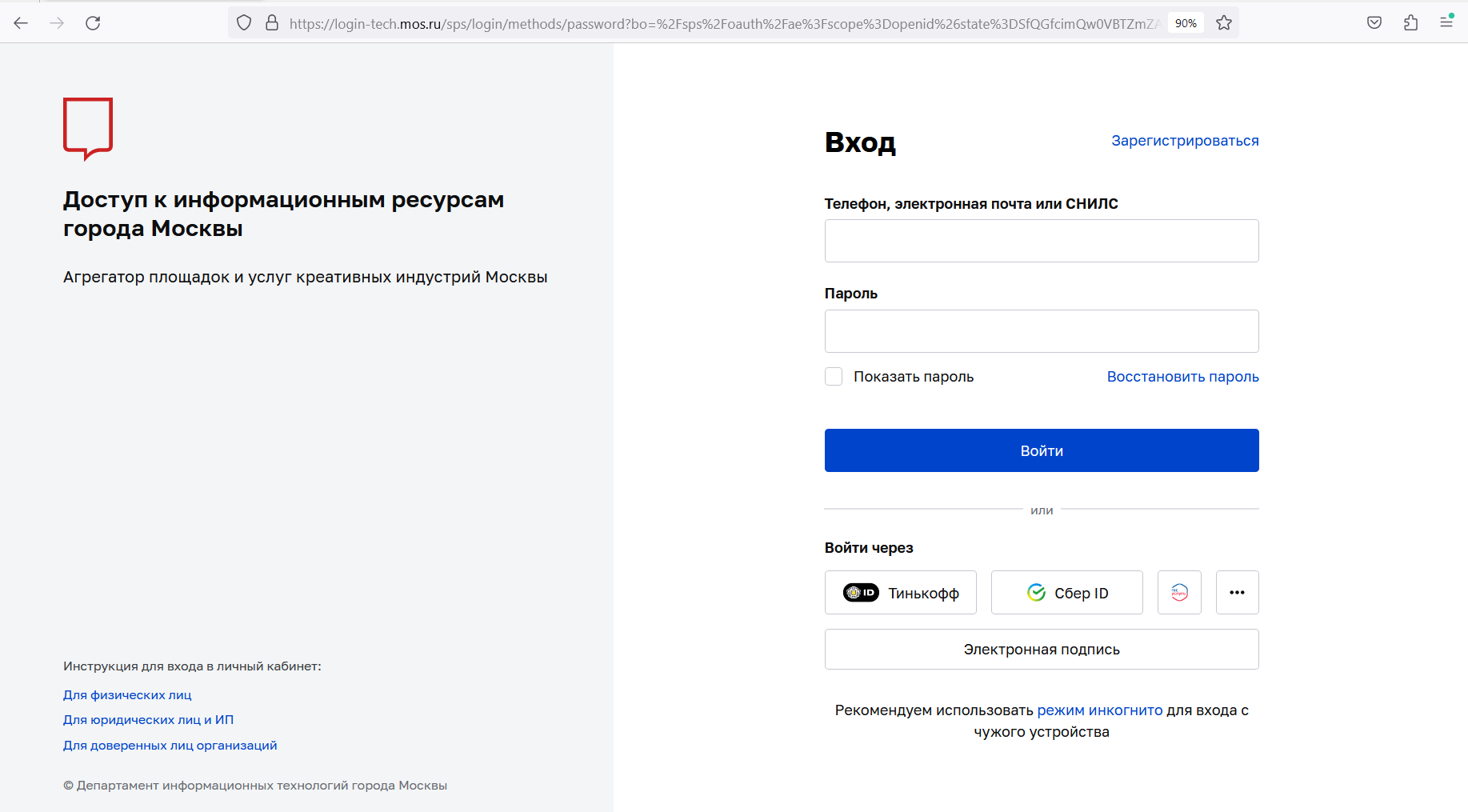
****

Рисунок 20. Форма аутентификации **СУДИР**

#### Изменение дизайна

Доработан дизайн формы аутентификации пользователей Платформы

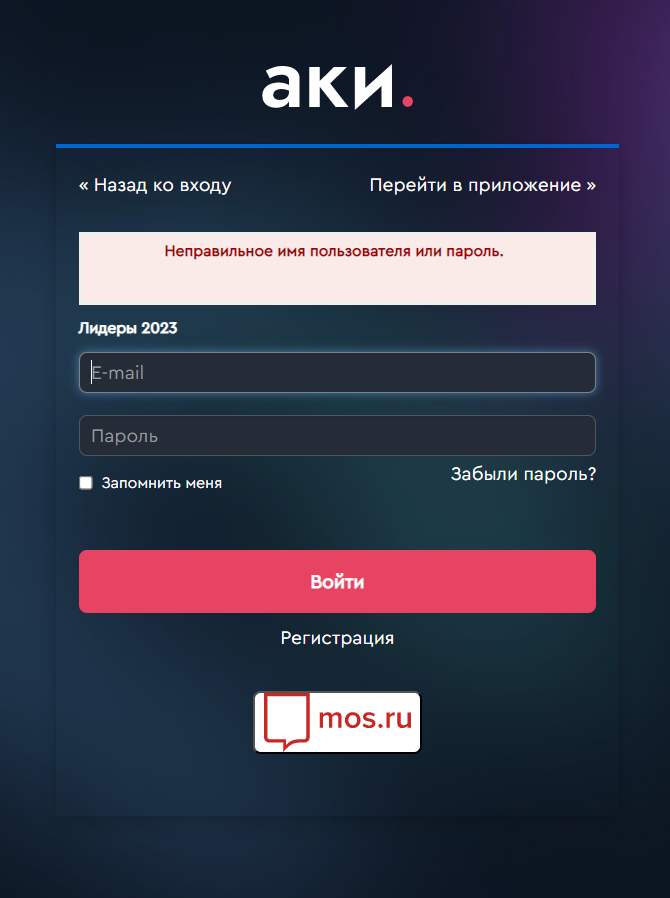


Рисунок 21. Форма аутентификации

Доработан дизайн формы аутентификации пользователей Платформы

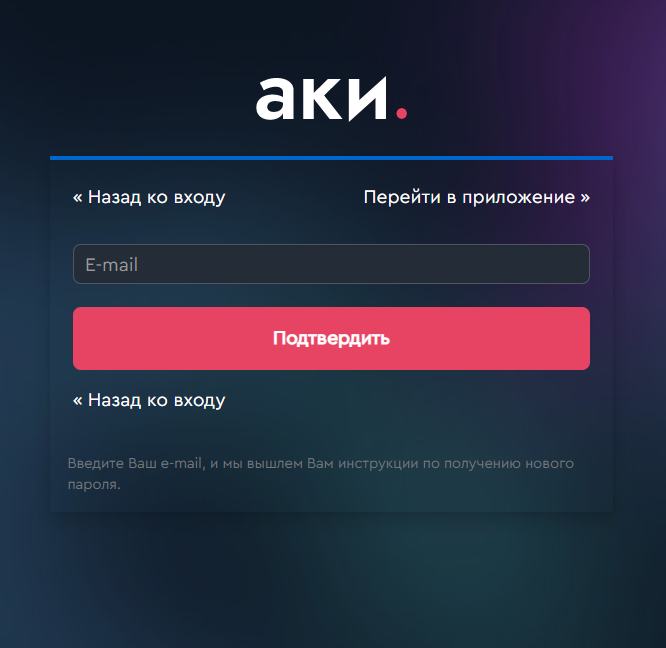


Рисунок 22. Восстановление пароля

Доработан дизайн формы регистрации пользователей Платформы

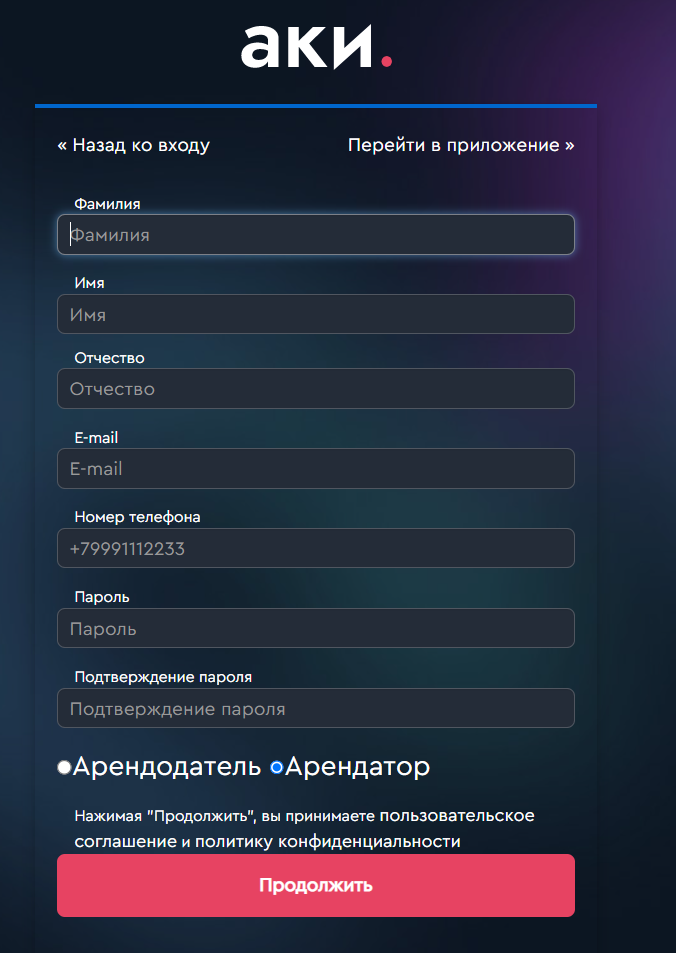


Рисунок 23. Самостоятельная регистрация Арендатора или Арендодателя

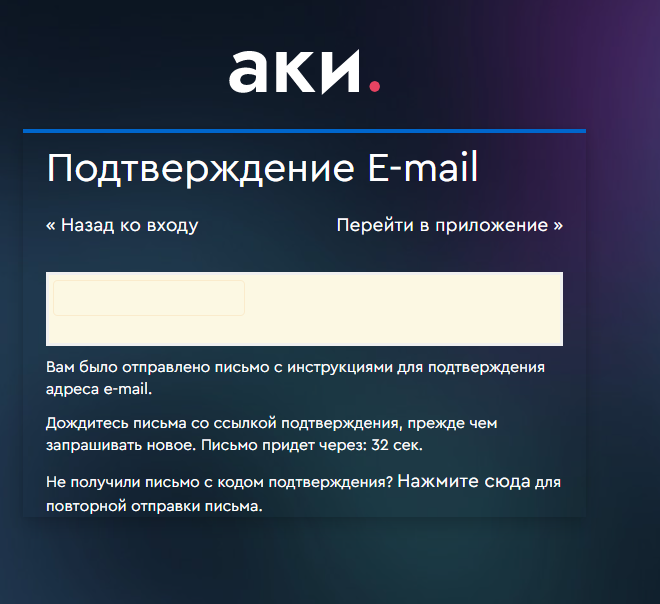


Рисунок 24. Самостоятельная регистрация Арендатора или Арендодателя (подтверждение email)

Доработан дизайн формы аутентификации администратора сервера OAuth

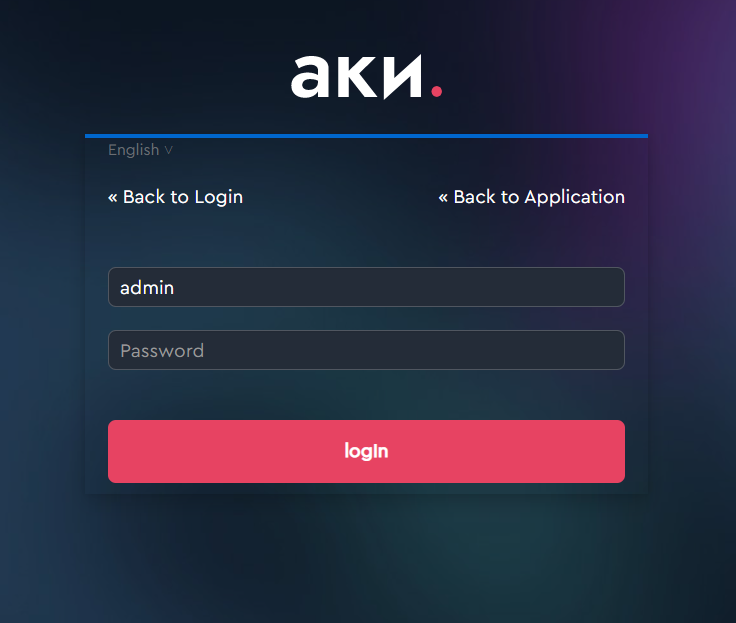


Рисунок 25. Аутентификация администратора (управление пользователями)

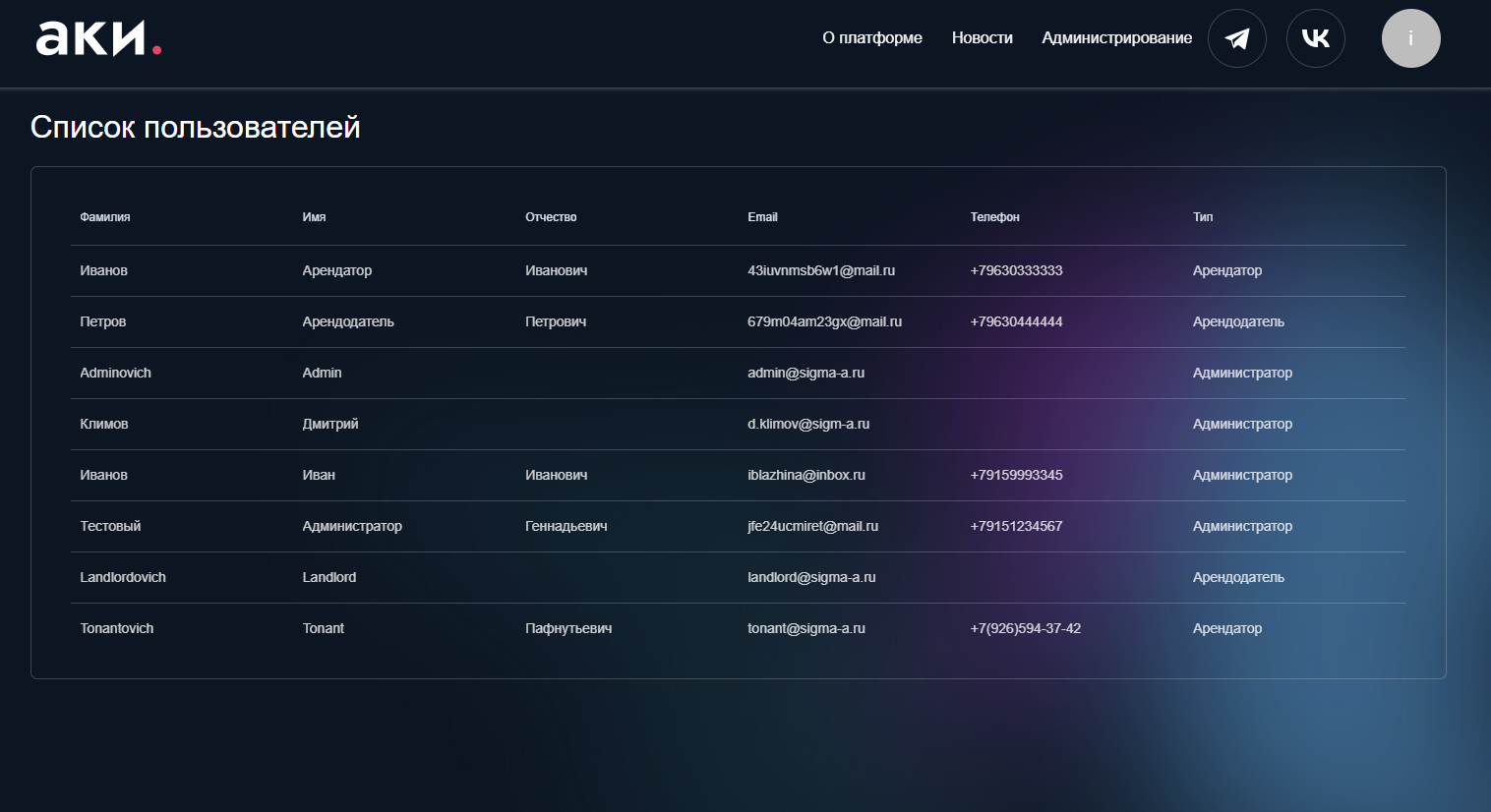


Рисунок 26. Управление пользователями администратором через платформу)