Staff

система управления персоналом

Документация программного обеспечения

Страница содержит сведения, необходимые для соответствия п. 4 «ж» Правил ведения реестра российского ПО (ПП РФ № 1236 от 16.11.2015).

1. Описание функциональных характеристик
2. Информация для установки
3. Информация для эксплуатации

РАЗДЕЛ 1

Описание функциональных характеристик

Назначение системы

Staff — веб-ориентированная информационная система управления персоналом, разработанная для автоматизации HR-процессов организации. Система обеспечивает ведение базы сотрудников, управление вакансиями и кандидатами, учёт стажёров, а также интеграцию со смежными сервисами через REST API.

Система реализована как SaaS-платформа, развёртываемая в инфраструктуре Kubernetes. Работа осуществляется через веб-интерфейс без установки дополнительного ПО на стороне пользователя.

Тип решения	Self-hosted (Kubernetes / Helm)
Интерфейс пользователя	Веб-браузер, без локальной установки
Режим работы	24/7
Мобильное приложение	Не предусмотрено

Функциональные модули

База сотрудников Ведение профилей сотрудников, должностей, расписаний и кадровой истории	**Вакансии и найм** Управление вакансиями, кандидатами и процессом проведения интервью
Учёт стажёров Отдельный контур управления стажёрами и их трудовой деятельностью	**REST API** Открытый API для интеграций со смежными сервисами: notifier, loyalty-program, dashboard

Технологический стек

Компонент	Технология	Роль
Веб-интерфейс	React 18, TypeScript 5.0, Vite 6	Административный интерфейс HR
Серверная часть	NestJS 9, TypeScript 4.7, Node.js 22	Бизнес-логика, данные, интеграции
База данных	PostgreSQL 15	Основное хранилище (139 миграций)
Файловое хранилище	S3 / MinIO	Документы и файлы сотрудников
Брокер сообщений	RabbitMQ 3.12	Асинхронная обработка событий

API-документация

Интерактивная документация REST API (Swagger / OpenAPI) доступна в развёрнутом экземпляре по адресу /api/docs. Документация защищена Basic Auth и доступна в рамках предоставленного тестового контура.

РАЗДЕЛ 2

Информация для установки

Способ развёртывания

Staff развёртывается как набор контейнеров в кластере Kubernetes с использованием Helm. Для доступа к системе локальная установка на стороне пользователя не требуется — достаточно современного браузера.

> **i** Для проведения экспертной проверки предоставляется готовый тестовый контур. Локальное развёртывание для рядового пользователя не требуется.

Системные требования

Компонент	Требование
Браузер (пользователь)	Chrome, Firefox, Edge, Safari актуальных версий
Интернет-соединение	Требуется для доступа к развёрнутому экземпляру
Node.js (разработчик)	18 и выше
Пакетные менеджеры	pnpm 10+ (backend), yarn 1.22+ (frontend)
Контейнеризация	Docker CE, docker-compose
СУБД	PostgreSQL 15
Брокер	RabbitMQ 3.12
Оркестрация	Kubernetes, Helm 3.14+

Порядок получения доступа к экземпляру

1	Запрос тестового аккаунта Ответственный сотрудник команды Staff формирует отдельный тестовый контур и учётную запись с ролью admin.
2	Получение учётных данных Эксперту направляются URL тестового экземпляра, логин и пароль администратора.
3	Вход в систему Открыть URL в браузере. На странице входа ввести логин и пароль. Система автоматически определяет роль и открывает доступ к разделам.

4	Доступ к инфраструктуре (при необходимости) Для просмотра ресурсов Kubernetes предоставляется kubeconfig с правами на чтение в namespace staff-box. Проверка: kubectl get pods -n staff-box

Параметры тестового контура

Адрес	https://staff.box.flang.io
Инфраструктура	Kubernetes, Yandex Cloud
Namespace	staff-box
Роль	admin — полный доступ ко всем разделам

Хранение исходного кода

Исходный код хранится в self-hosted GitLab, развёрнутом в Yandex Cloud. Репозиторий реализован как монорепозиторий: директория backend/ содержит NestJS REST API, директория frontend/ — React 18 SPA. Сборка и деплой производятся через GitLab CI/CD (стадии: build → deploy) с использованием Helm upgrade в Kubernetes.

Ветка	Назначение
main	Стабильная продуктивная ветка
develop	Основная ветка разработки
feat/*	Функциональные ветки
fix/*	Ветки исправлений

РАЗДЕЛ 3

Информация для эксплуатации

Режим работы

Система функционирует в режиме 24/7. Обновления выполняются через GitLab CI/CD (Helm upgrade) без остановки сервиса. Резервное копирование базы данных PostgreSQL обеспечивается средствами инфраструктуры Yandex Cloud.

Техническая поддержка — SLA

Тип инцидента	Время реакции	Приоритет
Критические инциденты (блокирующие)	до 15 минут	критично
Высокий приоритет	до 2 часов	высокий
Средний и низкий приоритет	до 24 часов	стандарт

Каналы обращения: Telegram, e-mail. Контакты предоставляются вместе с учётными данными.

Процессы обработки инцидентов

- Приём и регистрация обращения клиентским менеджером (первая линия поддержки)
- Классификация по критичности: блокирующие, высокие, средние, низкие
- Эскалация в команду разработки при необходимости
- Устранение: багфиксы, восстановление данных, настройка интеграций
- Информирование пользователя об устранении инцидента

Развитие и обновления

- Регулярные релизы каждые 1–3 недели с новыми функциями и исправлениями
- Документирование изменений (release notes) для каждого релиза
- Автоматизированное тестирование: юнит-, e2e-, smoke- и regression-тесты в CI/CD
- Мониторинг метрик производительности и ошибок в production-контуре
- Поддержка и развитие REST API для интеграций со смежными сервисами

Состав команды сопровождения

Роль	Зона ответственности
Клиентский менеджер	Первая линия поддержки, приём и регистрация обращений, консультирование
Проектный менеджер	Бэклог, приоритизация задач, баланс развития и устранения проблем
Backend-разработчик	Серверная часть, REST API, миграции базы данных

Frontend-разработчик	Веб-интерфейс, визуальные и функциональные исправления
QA-инженер	Тестирование функций и багфиксов, smoke/regression перед релизом
DevOps-инженер	CI/CD, мониторинг инфраструктуры, резервное копирование, Kubernetes

---

Документ подготовлен для соответствия требованиям п. 4 «ж» Правил формирования и ведения единого реестра российских программ для ЭВМ (ПП РФ № 1236 от 16.11.2015). Flang Platform.