

Система электронного учета заявлений

Электронная система учёта заявлений — это внутренняя платформа для сотрудников, руководителей и администраторов компании, обеспечивающая цифровое оформление, автоматическую маршрутизацию и контроль заявлений (отпуск, командировка, .больничный и др.)



Зачем нужна система?

- Упростить процесс подачи и согласования заявлений
- Уйти от бумажной волокиты
- Обеспечить прозрачность и контроль
- Автоматизировать рутинные задачи руководителей и HR
- Обеспечить единый архив

Основные функции для сотрудников

Создание заявлений

Сотрудники могут быстро создавать заявления по шаблонам (отпуск, командировка, расходы), заполняя персональные данные.

История заявлений

Возможность просматривать историю всех заявлений для контроля статуса и решений.

Прикрепление документов

Поддержка загрузки файлов (pdf, word, docx) для предоставления дополнительной информации.

Уведомления

Автоматические email-уведомления о статусе заявлений для своевременного информирования.

Ролевая модель и пользовательские требования

Роль	Функции	Пример
Пользователь (сотрудник)	Создание заявлений, просмотр статусов, дополнение заявлений	Авторизация, выбор шаблона, прикрепление документов, просмотр результата
Руководитель	Управление заявлениями команды, одобрение, комментарии	Обработка заявлений, изменение статусов, уведомления
Администратор	Управление шаблонами, пользователями, мониторинг системы	Редактирование шаблонов, управление ролями, анализ отчетов

Жизненный цикл заявления

1. **Сотрудник** выбирает шаблон → заполняет → прикрепляет документы → отправляет.
2. **Система** направляет заявление руководителю и уведомляет его.
3. **Руководитель** просматривает заявление, принимает решение (одобряет/отклоняет) и может добавить комментарий.
4. **Сотрудник** получает уведомление о результате.
5. **Администратор** управляет шаблонами, пользователями и может мониторить систему.

С точки зрения архитектуры

Система представляет собой внутреннее корпоративное веб приложение для учета заявлений сотрудников. Ключевые аспекты безопасность, масштабируемость, веб-интерфейс, роли (сотрудник, руководитель, администратор), работа с шаблонами, документооборот и уведомления.

Архитектура приложения: монолитное веб приложение, развертывание в корпоративной сети (без доступа из сети интернет), JWT для аутентификации, отсутствие микросервисов

Технологический стек

Предположительные технологии для реализации

Backend: язык Python (фреймворк Flask или FastApi)

Аутентификация: JWT (через pyjwt, fastapi-jwt-auth)

Работа с базой данных: ORM SQLAlchemy (Python)

Хранение файлов: локальная файловая система + ограничение по MIME типу и весу

Email-уведомления: SMTP через smtplib

База данных: PostgreSQL

Frontend: React + Typescript