

# Проект: RemoteDesk (рабочее название)

---

## 1. Общая информация

### 1.1 Цель проекта

Разработать кроссплатформенную систему удалённого доступа к компьютерам через интернет с возможностью:

- просмотра экрана удалённого устройства;
- управления мышью и клавиатурой;
- передачи файлов;
- организации защищённого соединения;
- управления списком подключённых устройств.

### 1.2 Целевая аудитория

- IT-специалисты
  - Техподдержка
  - Компании с удалёнными сотрудниками
  - Частные пользователи
- 

## 2. Архитектура системы

### 2.1 Общая схема

**Компоненты:**

1. Web-клиент (Angular)
  2. Desktop-агент (Windows / Linux / macOS)
  3. Signaling Server (WebRTC)
  4. FTP-сервер (передача файлов)
  5. Backend API (Node.js / NestJS)
-

## 2.2 Технологический стек

### Frontend:

- Angular 17+
- TypeScript
- RxJS
- WebRTC API
- Angular Material

### Backend:

- Node.js
- NestJS
- WebSocket (для сигналинга)
- JWT-авторизация
- PostgreSQL

### Desktop Agent:

- Electron или C++ / Rust
- WebRTC
- Системные API для захвата экрана
- Инжектирование ввода (mouse/keyboard)

### Передача файлов:

- FTP (через защищённый канал FTPS)
- 

## 3. Функциональные требования

---

### 3.1 Авторизация и регистрация

#### Пользователь может:

- Зарегистрироваться
- Войти в систему
- Восстановить пароль
- Подключить 2FA (опционально)

## **Безопасность:**

- JWT access + refresh токены
  - Шифрование паролей (bcrypt)
  - HTTPS обязателен
- 

## **3.2 Управление устройствами**

### **Пользователь может:**

- Добавить устройство
- Удалить устройство
- Видеть онлайн / оффлайн статус
- Назначать имя устройству

### **Устройство получает:**

- Уникальный ID
  - Пару ключей (public/private)
- 

## **3.3 Удалённое подключение**

### **3.3.1 Установление соединения**

1. Клиент инициирует подключение.
  2. Через Signaling Server происходит обмен SDP.
  3. Устанавливается P2P WebRTC соединение.
  4. В случае NAT используется TURN сервер.
- 

### **3.3.2 Функции удалённого доступа**

- Просмотр экрана в реальном времени
  - Управление мышью
  - Управление клавиатурой
  - Отображение курсора
  - Поддержка нескольких мониторов
-

### **3.3.3 Требования к качеству**

- Задержка не более 200 мс
  - Поддержка 720p / 1080p
  - Адаптивный bitrate
- 

## **3.4 Передача файлов**

Через FTP:

Функции:

- Отправка файла
- Загрузка файла
- Отображение прогресса
- Ограничение размера (по настройке)

Передача должна быть:

- Зашифрованной (FTPS)
  - Авторизованной через токен
- 

## **3.5 Администрирование**

Администратор может:

- Блокировать пользователей
  - Видеть активные сессии
  - Логировать подключения
  - Принудительно завершать сессию
- 

## **4. Нефункциональные требования**

**Производительность:**

- Поддержка 10 000 одновременных подключений (горизонтальное масштабирование)

### **Безопасность:**

- End-to-End шифрование WebRTC
- Защита от brute-force
- Ограничение частоты запросов
- IP-логирование

### **Логирование:**

- Все сессии сохраняются
  - История подключений доступна пользователю
- 

## **5. UI/UX требования**

### **Главный экран:**

- Список устройств
- Кнопка подключения
- Статус устройства

### **Экран сессии:**

- Видео поток
  - Панель управления
  - Кнопка передачи файла
  - Кнопка завершения сессии
- 

## **6. Этапы разработки**

### **Этап 1**

- Авторизация
- База данных
- Добавление устройств

## **Этап 2**

- Реализация Signaling
- WebRTC соединение
- Захват экрана

## **Этап 3**

- Управление мышью и клавиатурой
- Передача файлов

## **Этап 4**

- Админка
  - Логирование
  - Тестирование
- 

## **7. Критерии приёмки**

Проект считается завершённым если:

- Устанавливается стабильное P2P соединение
- Работает передача видео
- Работает удалённое управление
- Передаются файлы
- Система выдерживает нагрузочное тестирование