# Техническое задание / Technical Specification

## Русская версия

### 1. Общие требования

Стек технологий:  
- Python 3  
- Django + Django REST Framework  
- PostgreSQL (рекомендуется) или MySQL  
- Установка на домашнем сервере  
- Возможность использования Docker и Nginx  
  
Развёртывание:  
- Работа в локальной сети и через внешний доступ  
- Возможность резервного копирования базы данных

### 2. Модели данных

Пользователь (User): id, username, password, email, phone, role, created\_at, updated\_at

Заказ (Order): id, client\_name, client\_phone, client\_address, device\_model, problem\_description, status, assigned\_to, price\_estimate, final\_price, created\_at, updated\_at, deleted

Лента событий (OrderEvent): id, order\_id, user\_id, event\_type, event\_description, created\_at

### 3. Функциональные требования

Пользователи: регистрация, авторизация, редактирование профиля, смена пароля, удаление аккаунта.

Заказы: создание, просмотр списка с фильтрацией, обновление, удаление (soft delete).

Лента событий: автоматическая запись изменений по заказу, просмотр истории событий.

### 4. Права доступа

Администратор: полный доступ.  
Диспетчер: управление заказами, назначение мастеров.  
Мастер: работа только со своими заказами.

### 5. Безопасность

JWT Token Authentication, валидация данных, разграничение прав доступа.

### 6. Тестирование

Unit-тесты для CRUD-операций, интеграционные тесты для заказов и событий.

### 7. Документация

README с инструкциями, API-документация (Swagger/OpenAPI).

### 8. User Flow

Администратор: управление пользователями и заказами.

Диспетчер: создание и назначение заказов.

Мастер: просмотр и изменение статуса назначенных заказов, добавление комментариев.

### 9. Дополнительно

Экспорт заказов в Excel/CSV, резервное копирование базы данных, HTTPS-доступ.

## English Version

### 1. General Requirements

Technology stack:  
- Python 3  
- Django + Django REST Framework  
- PostgreSQL (recommended) or MySQL  
- Deployment on a home server  
- Optional: Docker and Nginx  
  
Deployment:  
- Accessible within local network and (optionally) external access  
- Database backup support

### 2. Data Models

User: id, username, password, email, phone, role, created\_at, updated\_at

Order: id, client\_name, client\_phone, client\_address, device\_model, problem\_description, status, assigned\_to, price\_estimate, final\_price, created\_at, updated\_at, deleted

OrderEvent: id, order\_id, user\_id, event\_type, event\_description, created\_at

### 3. Functional Requirements

Users: registration, login, profile view/update, password change, account deletion.

Orders: create, retrieve with filtering, update, delete (soft delete).

Order Events: automatic log of changes per order, view event history.

### 4. Roles & Permissions

Administrator: full access.  
Dispatcher: manage orders, assign technicians.  
Technician: manage only assigned orders.

### 5. Security

JWT Token Authentication, data validation, role-based access control.

### 6. Testing

Unit tests for CRUD operations, integration tests for orders and events.

### 7. Documentation

README with setup instructions, API documentation (Swagger/OpenAPI).

### 8. User Flow

Administrator: manage users and orders.

Dispatcher: create and assign orders.

Technician: view and update assigned orders, add comments.

### 9. Additional Features

Export orders to Excel/CSV, database backup automation, HTTPS external access.