# Требования к ПО

## 1. Назначение ПО

Продукт предназначен для:

* Сборка и обработки информации о городских проблемах от жителей через мобильное приложение и веб-интерфейс.
* Администрирования заявок от пользователей с возможностью редактирования и изменения их статуса.
* Повышения прозрачности и скорости реагирования городских служб на проблемы.

## 2. Пользователи системы

* **Горожане (Пользователи)**:
  + Регистрация и авторизация.
  + Сообщение о проблемах с приложением фото и адреса.
  + Просмотр статуса отправленных проблем.
* **Администраторы городских служб (Админы)**:
  + Обработка и изменение статуса проблем.
  + Доступ к фильтрации и аналитике проблем.

## 3. Основные компоненты системы

### 3.1 Пользовательское ПО

* Регистрация / авторизация пользователя.
* Создание заявки о проблеме:
  + Выбор типа проблему из предложенных
  + Краткое описание проблемы.
  + Подробное описание
  + Прикрепление фотографии (до 5 МБ).
  + Автоматическое определение адреса по координатам или ручной ввод.
* Просмотр статуса своих проблем.
* Рейтинг?
* Интеграция с API через HTTPS.

### 3.2 Административная панель

* Просмотр зарегистрированных пользователей.
* Управление статусами проблем (новая, в обработке, решена).
* Фильтрация проблем (по типу, статусу, времени).
* Статистика( по типу, времени и адресу)

### 3.4 API-сервер на FastAPI

* Регистрация и авторизация с хешированием паролей.
* Генерация и проверка токенов подтверждения email.
* Приём и обработка заявок о проблемах.
* Обновление статуса проблем.
* Выдача краткой информации о проблемах через кэш-фрагменты.
* Фильтрация заявок через SQL или ORM-запросы.

### 3.5 Сервер и маршрутизация (NGINX)

* Реверс-прокси для FastAPI-сервера.
* Раздача статики (HTML, CSS, JS).
* Обеспечение SSL/TLS для всех соединений.
* Ограничение размеров загружаемых файлов (до 5 МБ).

### 3.6 База данных (PostgreSQL)

* **Таблица пользователей**:
  + id
  + логин
  + пароль (хешированный)
  + email
  + admin (boolean)
* **Таблица проблем**:
  + id
  + тип (категория проблемы)
  + краткое описание(до 200 символов)
  + подробное описание( до 1000)
  + статус
  + дата и время
  + адрес

## 4. Функциональные требования

| **№** | **Требование** | **Описание** |
| --- | --- | --- |
| 1 | Регистрация пользователя | Проверка данных, хеширование пароля |
| 2 | Аутентификация | Логин и пароль, проверка токена |
| 3 | Загрузка фотографии | Проверка размера и метаданных |
| 4 | Определение местоположения | По координатам или введённому адресу |
| 5 | Создание заявки на проблему | Фото, адрес, комментарий, тип |
| 6 | Просмотр статуса проблем |  |
| 7 | Изменение статуса проблемы | Изменение через панель администратора |
| 8 | Фильтрация проблем | По статусу, типу, дате |
| 9 | Рейтинг пользователей | Меняется в зависимости от правдивости информации |
| 10 | Подтверждение email | Генерация ссылки с токеном для подтверждения |
| 11 | Уведомление email |  |

## 5. Нефункциональные требования

* **Безопасность**:
  + Использование SSL-сертификатов для всех соединений.
  + Хранение паролей в зашифрованном виде.
  + Защита от SQL-инъекций (использование ORM или подготовленных запросов).
* **Производительность**:
  + Время отклика API не более 3с.
  + Поддержка до 50 пользователей одновременно.
  + Кэширование краткой информации о проблемах.
* **Надёжность**:
  + Обработка ошибок API с выдачей корректных ответов.
  + Логирование всех действий пользователей и администраторов.

## 6. Тестирование