**Мобильное приложение «ПИТТАСИ»**

**Руководство программиста**

Киров, 2025 г.

**Аннотация**

Настоящее руководство программиста предназначено для специалистов, осуществляющих разработку, сопровождение и модификацию информационной системы «ПИТТАСИ».

Документ содержит сведения о структуре программного комплекса, архитектурных решениях, технологическом стеке, правилах настройки окружения, сборки и развертывания компонентов системы.

Руководство программиста разработано с учётом требований ГОСТ 19.504–79 и ГОСТ Р 59795–2021.

**Содержание**

[Назначение и условия применения программы](#_Toc212601949)

[1.1 Назначение и функции программы 2](#_Toc212601950)

[1.2 Условия выполнения программы 2](#_Toc212601951)

[1.2.1 Требования к техническому обеспечению 2](#_Toc212601952)

[1.2.2 Требования к программному обеспечению 3](#_Toc212601953)

[1.2.3 Требования к информационному обеспечению 3](#_Toc212601954)

[Характеристика программы](#_Toc212601955)

[2.1 Описание основных характеристик 5](#_Toc212601956)

[2.2 Режимы работы программы 5](#_Toc212601957)

[2.3 Временные характеристики 6](#_Toc212601958)

[2.4 Средства контроля и самовосстанавливаемости 6](#_Toc212601959)

[Обращение к программе](#_Toc212601960)

[3.1 Процедура запуска системы 7](#_Toc212601961)

[3.2 Передача управления и параметров 7](#_Toc212601962)

[Входные и выходные данные](#_Toc212601963)

[4.1 Организация входной информации 9](#_Toc212601964)

[4.2 Организация выходной информации 9](#_Toc212601965)

[Сообщения 10](#_Toc212601966)

[5.1 Системные сообщения 10](#_Toc212601967)

[5.2 Сообщения об ошибках 10](#_Toc212601968)

[5.3 Действия по сообщениям 11](#_Toc212601969)

# Назначение и условия применения программы

В данном разделе описаны назначение, основные функции и условия применения информационной системы «ПИТТАСИ».

## Назначение и функции программы

Информационная система «ПИТТАСИ» представляет собой программный комплекс с клиент-серверной архитектурой, предназначенный для автоматизации процесса взаимодействия между жителями города и муниципальными службами.

Система обеспечивает выполнение следующих основных функций:

* аутентификация и авторизация пользователей с разделением ролей («Житель города», «Модератор»);
* создание, редактирование и просмотр обращений о проблемах городской инфраструктуры с прикреплением фотоматериалов и геолокационной метки;
* визуализация всех обращений на интерактивной карте;
* реализация социальных механик: публичное комментирование обращений и голосование за их приоритетность;
* модерация обращений: изменение статусов, добавление официальных ответов;
* предоставление публичного API для доступа к обезличенным статистическим данным.

## Условия выполнения программы

В настоящем подразделе описаны условия, необходимые для корректной разработки, развертывания и функционирования системы «ПИТТАСИ».

### Требования к техническому обеспечению

Для разработки и развертывания системы требуется рабочая станция или сервер, отвечающий следующим минимальным требованиям:

* операционная система: Windows 10/11 с WSL2, macOS 11.0 и выше, или дистрибутив Linux (например, Ubuntu 22.04 LTS);
* оперативная память: не менее 8 ГБ (рекомендуется 16 ГБ для комфортной работы с эмуляторами и Docker);
* свободное место на жёстком диске: не менее 20 ГБ для исходного кода, SDK, Docker-образов и баз данных;
* стабильное подключение к сети Интернет.

### Требования к программному обеспечению

Для успешной разработки, сборки и функционирования системы «ПИТТАСИ» используются следующие программные средства:

**Серверная часть (Backend):**

* среда выполнения JavaScript: Node.js (версия 18.x LTS или выше);
* фреймворк: NestJS (версия 9.x или выше) на языке TypeScript;
* API: GraphQL, реализованный с помощью библиотеки Apollo Server;
* ORM (Object-Relational Mapping): Prisma для взаимодействия с базой данных;
* система управления базами данных: PostgreSQL (версия 15 или выше);
* расширение СУБД: PostGIS для работы с геопространственными данными.

**Клиентская часть (Мобильное приложение):**

* SDK: Dart SDK (версия 3.x или выше);
* фреймворк: Flutter (версия 3.x или выше) для кроссплатформенной разработки;
* картографический сервис: Yandex MapKit.

**Общие инструменты разработки и развертывания:**

* система контейнеризации: Docker и Docker Compose;
* система контроля версий: Git;
* среда разработки: Visual Studio Code, Android Studio (для Android), Xcode (для iOS).

### Требования к информационному обеспечению

Информационное обеспечение системы реализуется на основе реляционной базы данных под управлением СУБД PostgreSQL с расширением PostGIS.

Все данные организованы в логически взаимосвязанные таблицы, обеспечивающие целостность и непротиворечивость данных с помощью механизмов первичных и внешних ключей.

Основные сущности БД:

* пользователи (Users): данные о зарегистрированных пользователях, включая их роли (житель, модератор);
* обращения (Problems/Issues): основная сущность, содержащая описание проблемы, категорию, статус, координаты (геопространственный тип данных), а также ссылки на автора и прикрепленные файлы;
* категории (Categories): справочник категорий проблем (например, «Дороги», «Освещение»);
* комментарии (Comments): тексты комментариев, привязанные к конкретным обращениям и пользователям;
* голоса (Votes): записи о голосах пользователей за приоритетность обращений.

Для всех внешних ключей установлено правило каскадного удаления (ON DELETE CASCADE) для поддержания ссылочной целостности.

# Характеристика программы

В настоящем разделе приведено описание основных характеристик и особенностей информационной системы «ПИТТАСИ».

## Описание основных характеристик

Разрабатываемая система представляет собой программный комплекс с клиент-серверной архитектурой. Мобильное приложение (клиент) разработано на кроссплатформенном фреймворке Flutter. Серверная часть (бэкенд) реализована на фреймворке NestJS.

Взаимодействие между клиентом и сервером осуществляется посредством GraphQL API, что позволяет мобильному приложению запрашивать только необходимые в данный момент данные, экономя трафик и ускоряя работу.

Система ориентирована на две основные категории пользователей с разными правами доступа:

Приложение ориентировано на две категории пользователей:

* Житель города — основной пользователь приложения, который может сообщать о проблемах и участвовать в их обсуждении;
* Модератор — представитель муниципальных служб, ответственный за обработку и изменение статусов обращений (управление осуществляется через отдельный веб-интерфейс, не рассматриваемый в данном руководстве).

## Режимы работы программы

Система поддерживает несколько режимов работы, определяемых ролью пользователя.

Режим «Житель города»:

* регистрация, аутентификация и управление собственным профилем;
* создание новых обращений с прикреплением фото и указанием геолокации;
* просмотр всех обращений на интерактивной карте;
* фильтрация обращений на карте по категориям и статусам;
* просмотр детальной информации по каждому обращению;
* голосование за приоритетность проблемы;
* добавление комментариев к обращениям;
* просмотр списка собственных обращений и отслеживание их статусов.

Режим «Модератор»:

* все функции, доступные «Жителю города»;
* изменение статуса любого обращения («Новое», «В работе», «Решено»);
* добавление официальных комментариев от лица службы.

Все режимы доступны при наличии подключения к сети Интернет.

## Временные характеристики

Система «ПИТТАСИ» обеспечивает стабильную работу при подключении к сети Интернет со скоростью не ниже 3 Мбит/с.

Показатели производительности:

* время отклика GraphQL API на стандартные запросы — не более 500 мс;
* загрузка главной страницы с картой и метками в видимой области — не более 3 секунд;
* создание нового обращения (без учета времени загрузки фото) — не более 2 секунд;
* время загрузки фото зависит от скорости интернет-соединения пользователя.

Время восстановления работоспособности системы после критического сбоя серверной части составляет не более 2 часов, включая процедуры восстановления контейнеров и базы данных PostgreSQL из резервных копий.

## Средства контроля и самовосстанавливаемости

Система реализует комплексный подход к контролю целостности данных и самовосстановлению.

**Контроль целостности данных обеспечивается за счёт:**

* двухуровневой валидации вводимой информации: на стороне клиента (средствами Flutter) и на стороне сервера (средствами NestJS Pipes и Prisma);
* строгой типизации данных, обеспечиваемой TypeScript на бэкенде и Dart на клиенте;
* обеспечения ссылочной целостности в базе данных PostgreSQL с помощью внешних ключей и каскадных операций.

**Механизмы самовосстановления системы включают:**

* регулярное резервное копирование базы данных с помощью встроенных CRON-задач в среде Docker;
* автоматический перезапуск контейнеров Docker (restart: always) при возникновении сбоев в работе сервисов (NestJS, PostgreSQL);
* систему логирования и мониторинга ошибок на сервере через встроенный логгер NestJS для своевременного обнаружения и диагностики проблем.

# Обращение к программе

В настоящем разделе описаны процедуры запуска, инициализации и взаимодействия с информационной системой «ПИТТАСИ».

## Процедура запуска системы

Развертывание и запуск серверной части системы осуществляется в среде Docker. Для начала работы необходимо выполнить следующие действия:

* клонировать репозиторий проекта с официального источника (например, GitHub);
* перейти в корневую директорию серверной части проекта;
* создать конфигурационный файл .env на основе файла-примера .env.example и указать в нем необходимые переменные окружения (строка подключения к БД, секретный ключ для JWT и т.д.);
* запустить контейнеры приложения командой: docker-compose up -d.

После успешного развертывания серверная часть будет доступна для приема запросов от мобильного приложения. Запуск клиентской части осуществляется путем сборки и установки мобильного приложения на целевое устройство или эмулятор.

## Передача управления и параметров

Управление системой выполняется через мобильное приложение, которое взаимодействует с сервером посредством GraphQL API.

Передача параметров между клиентом и сервером осуществляется с помощью стандартных операций GraphQL:

* Queries: для получения данных (список обращений, информация о пользователе и т.д.);
* Mutations: для создания и изменения данных (создание обращения, добавление комментария, голосование);
* Subscriptions: для организации real-time обновлений (например, мгновенное обновление статуса обращения).
* Пользователь взаимодействует с системой через:
* формы ввода данных (регистрация, создание обращения);
* интерактивную карту (выбор местоположения, просмотр меток);
* кнопки действий («Отправить», «Проголосовать», «Выйти»).

# Входные и выходные данные

В данном разделе описана организация используемой входной и выходной информации информационной системы «ПИТТАСИ».

## Организация входной информации

Входная информация поступает в систему от пользователей через мобильное приложение. Она включает несколько категорий данных в зависимости от выполняемых действий.

Основные категории входных данных:

* учетные данные пользователей: логин, email, пароль, используемые при регистрации и авторизации;
* данные профиля пользователя: имя, новый пароль, аватар;
* данные об обращении: координаты, категория, текстовое описание, прикрепленные фотоматериалы (в формате .jpg, .png);
* данные о взаимодействиях: идентификатор обращения для голосования или комментирования, текст комментария.

Все входные данные проходят многоуровневую проверку:

* валидация форм и форматов файлов на клиенте (Flutter);
* серверная проверка типов данных, длины строк и бизнес-логики через DTO (Data Transfer Objects) в NestJS и типизированную схему GraphQL.

## Организация выходной информации

Выходная информация формируется системой в зависимости от выполненных операций и роли пользователя.

Основные виды выходных данных:

* данные для мобильного приложения: информация, передаваемая в ответ на GraphQL-запросы в формате JSON (списки обращений, данные профиля, результаты голосования и т.д.);
* системные уведомления: push-уведомления об изменении статуса обращения, отправляемые на устройство пользователя;
* данные публичного API: обезличенные статистические данные, доступные по отдельным GraphQL-запросам для внешнего анализа;
* служебная информация: логи веб-приложения (ошибки, системные события, обращения к API), метрики производительности.

# Сообщения

В данном разделе описаны системные и пользовательские сообщения, выдаваемые в ходе работы мобильного приложения «ПИТТАСИ», а также рекомендации по действиям при их получении.

## Системные сообщения

Система информирует пользователя о результатах выполнения операций посредством всплывающих уведомлений (toast-сообщений), push-уведомлений и динамических изменений интерфейса.

Основные случаи отображения системных сообщений:

* при успешной авторизации: пользователь автоматически перенаправляется на главный экран с картой, соответствующий его роли;
* при создании обращения: отображается уведомление «Обращение успешно создано»;
* при успешном голосовании: счетчик голосов у обращения инкрементируется, а иконка голосования меняет свое состояние (например, становится неактивной);
* при добавлении комментария: новый комментарий немедленно появляется в списке под обращением;
* при изменении статуса обращения (модератором): автору обращения отправляется push-уведомление с новым статусом;
* при обновлении профиля: отображается сообщение «Изменения успешно сохранены».

Изменения состояния данных (например, появление новой метки на карте после создания обращения) отражаются в интерфейсе динамически, без необходимости перезагрузки экрана.

## Сообщения об ошибках

Система классифицирует ошибки по категориям и отображает информативные уведомления с пояснениями.

|  |  |  |
| --- | --- | --- |
| **Ситуация** | **Сообщение** | **Причина** |
| Ошибка входа | «Введены неверный логин или пароль» | Неправильные учетные данные. |
| Ошибка регистрации | «Пользователь с таким email уже существует» | Нарушение уникальности email пользователя. |
| Ошибка создания обращения | «Не все обязательные поля заполнены» | Пропуск обязательных для заполнения полей формы (например, описания). |
| Ошибка загрузки фото | «Недопустимый формат файла. Разрешены JPG и PNG» или «Размер файла превышает 5 МБ» | Ошибка формата файла или превышение установленного лимита на размер. |
| Ошибка определения геолокации | «Не удалось определить геолокацию. Проверьте настройки GPS и разрешения для приложения» | Отключена служба геолокации на устройстве или пользователь не предоставил приложению доступ. |
| Ошибка доступа | «Недостаточно прав для выполнения операции» | Попытка пользователя выполнить действие, не соответствующее его роли (например, житель пытается изменить статус обращения). |
| Системная ошибка (ошибка сети) | «Произошла ошибка на сервере. Проверьте подключение к интернету и повторите попытку позже» | Потеря соединения с сервером, недоступность API, непредвиденный сбой на бэкенде. |

## Действия по сообщениям

Рекомендации пользователю при возникновении ошибок:

* при ошибке авторизации: рекомендовать пользователю проверить правильность введённого логина и пароля, при необходимости воспользоваться функцией восстановления пароля (если реализована);
* при ошибке загрузки файла: рекомендовать пользователю убедиться, что изображение соответствует требованиям формата (JPG, PNG) и не превышает лимит по размеру (5 МБ), затем повторить загрузку;
* при ошибках валидации формы: убедиться, что клиентская и серверная валидация работают корректно и все обязательные поля помечены в интерфейсе;
* при ошибке геолокации: рекомендовать пользователю проверить системные настройки устройства и выданные приложению разрешения;
* при системных ошибках или ошибках доступа: проанализировать логи серверного приложения (NestJS) и состояние Docker-контейнеров для выявления первопричины сбоя. При необходимости выполнить повторный запрос позже.