|  |  |  |
| --- | --- | --- |
|  |  | УТВЕРЖДАЮ |
|  |  | Преподаватель УП.03  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долженкова М.Л.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку информационной системы для фиксации и учета дефектов объектов городской инфраструктуры.

|  |  |  |
| --- | --- | --- |
|  |  | СОГЛАСОВАНО |
|  |  | Студент колледжа ВятГУ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шумилов И.А.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |
|  |  |
|  |  |
|  |  |
|  |  |

2025

Содержание

[Введение 2](#_Toc212572501)

[1 Термины и определения 3](#_Toc212572502)

[2 Перечень сокращений 4](#_Toc212572503)

[3 Основные сведения о разработке 5](#_Toc212572504)

[3.1 Наименование разработки 5](#_Toc212572505)

[3.2 Цель и задачи 5](#_Toc212572506)

[3.3 Сведения об участниках разработки 5](#_Toc212572507)

[3.4 Сроки разработки 5](#_Toc212572508)

[3.5 Назначение разработки 6](#_Toc212572509)

[3.5.1 Функциональное назначение 6](#_Toc212572510)

[3.5.2 Эксплуатационное назначение 6](#_Toc212572511)

[4 Описание предметной области 7](#_Toc212572512)

[5 Требования к результатам разработки 12](#_Toc212572516)

[5.1 Требования к функциональным характеристикам 12](#_Toc212572517)

[5.2 Требования к показателям назначения 12](#_Toc212572518)

[5.3 Требования к технологическому стеку 12](#_Toc212572519)

[5.4 Требования к пользовательскому интерфейсу 13](#_Toc212572520)

[5.5 Требования к видам обеспечения 19](#_Toc212572521)

[5.5.1 Требования к математическому обеспечению 19](#_Toc212572522)

[5.5.2 Требования к информационному обеспечению 19](#_Toc212572523)

[5.5.3 Требования к метрологическому обеспечению 19](#_Toc212572524)

[5.5.4 Требования к техническому обеспечению 19](#_Toc212572525)

[5.6 Требования к надежности 20](#_Toc212572526)

[5.7 Требования к безопасности 20](#_Toc212572527)

[5.8 Требования к патентной чистоте 21](#_Toc212572528)

[5.9 Требования к перспективам развития 21](#_Toc212572529)

[6 Состав и содержание работ 22](#_Toc212572530)

[7 Порядок разработки 23](#_Toc212572531)

[8 Требования к документированию 24](#_Toc212572532)

# Введение

В данном документе представлено техническое задание для разработки информационной системы «ПИТТАСИ», предназначенной для сбора, обработки и визуализации обращений граждан о проблемах городской инфраструктуры.

Данное техническое задание содержит: наименование разработки, цели и задачи, сведения об участниках разрабатываемого сайта, сроки разработки, описание предметной области и другие аспекты, и требования, которые помогут поэтапно создать сайт для исполнителя.

Техническое задание предназначено для заказчика с целью согласования результата разработки проекта на соответствие с ожиданиями заказчика.

# Термины и определения

Экранная форма – это форма, которая может включать в себя различные элементы: игровое поле, интерфейс, меню, информационные панели, анимации, текст и т.д.

Обращение — структурированная запись в базе данных системы, содержащая информацию о городской проблеме, включая ее геолокацию, категорию, текстовое описание, прикрепленные фотоматериалы и текущий статус исполнения.

API (Application Programming Interface) — программный интерфейс приложения; набор правил и инструментов, позволяющий различным программным компонентам взаимодействовать друг с другом.

Вайрфрейм (от англ. *wireframe* — каркас, сетка) — схематичное, низкодетализированное представление пользовательского интерфейса. Предназначено для демонстрации структуры экранов, расположения ключевых элементов (кнопки, поля ввода, навигация) и логики взаимодействия с пользователем без акцента на графическом дизайне.

Геоданные (пространственные данные) — данные о географическом местоположении объектов, представленные в виде координат (широта и долгота).

Рейтинг проблемы — числовой показатель, отражающий общественную значимость и приоритетность обращения. Формируется на основе количества голосов, отданных пользователями за конкретную проблему.

# Перечень сокращений

API (Application Programming Interface) – интерфейс прикладного программирования.

БД – база данных.

ИС – информационная система.

СУБД – система управления базами данных.

СТП – стандарт предприятия.

JWT (JSON Web Token) – веб-токен в формате JSON, используемый для аутентификации.

ORM (Object-Relational Mapping) – технология объектно-реляционного отображения, используемая для взаимодействия с базой данных.

UI (User Interface) – пользовательский интерфейс.

# Основные сведения о разработке

## Наименование разработки

Наименование данной разработки: информационная система «ПИТТАСИ» для сбора, обработки и визуализации обращений граждан о проблемах городской инфраструктуры.

## Цель и задачи

Целью разработки является реализация цифрового инструмента для эффективного взаимодействия между жителями города и ответственными службами, который обеспечивает прозрачность процесса обработки обращений и позволяет формировать общественный рейтинг приоритетности проблем.

Задачи разработки:

* проанализировать предметную область;
* выявить и сформулировать проблему;
* спроектировать архитектуру системы и базу данных;
* разработать пользовательские интерфейсы (мобильное приложение и веб-портал);
* реализовать серверную часть и API;
* разработать сопроводительную документацию;
* пройти приемо-сдаточные испытания.

## Сведения об участниках разработки

Заказчиком является преподаватель Колледжа ВятГУ в составе – Долженкова Мария Львовна – преподаватель УП.03;

Исполнителем является студент Колледжа ВятГУ группы ИСПк-403-52-00 – Шумилов Иван Андреевич.

## Сроки разработки

Разработка должна вестись в рамках следующих сроков:

* начало разработки: 18.09.2025
* конец разработки: 18.11.2025

## Назначение разработки

### Функциональное назначение

Основное функциональное назначение системы «ПИТТАСИ» заключается в предоставлении централизованной платформы для управления жизненным циклом обращений о проблемах городской инфраструктуры.

Ключевые функции:

* предоставление интерактивной карты для нанесения и просмотра меток о проблемах;
* создание обращений с прикреплением фотографий, текстового описания и выбора категории;
* обеспечение механизма голосования за проблемы для формирования общественного рейтинга и определения приоритетов;
* поддержка ролевой модели (Житель города, Модератор) с разграничением прав доступа и функций;
* предоставление веб-портала для модераторов для управления обращениями (изменение статуса, добавление официальных комментариев);
* автоматическая отправка уведомлений пользователям об изменении статуса их обращений;
* предоставление публичного API для доступа к обезличенным статистическим данным.

### Эксплуатационное назначение

Эксплуатационное назначение состоит в повышении эффективности совместного быта и улучшении командной динамики;

* повышение оперативности выявления и устранения городских проблем за счет упрощения процесса подачи обращений;
* увеличение прозрачности деятельности городских служб путем отслеживания статуса исполнения заявок;
* вовлечение граждан в процесс улучшения городской среды и усиление общественного контроля;
* предоставление инструмента для независимого анализа данных об эффективности работы городских служб.

# Описание предметной области

Предметной областью является управление жизненным циклом обращений граждан о проблемах городской инфраструктуры. Существующий процесс взаимодействия между жителями и муниципальными службами часто характеризуется высоким порогом входа для подачи заявки, непрозрачностью и отсутствием оперативной обратной связи. Это приводит к накоплению нерешенных проблем и снижению доверия к городским службам.

Информационная система «ПИТТАСИ» функционирует как цифровой мост, преобразующий неструктурированные жалобы в формализованные, геолоцированные обращения. Система внедряет объективный, основанный на данных подход к приоритизации задач за счет социальных механик, таких как общественное голосование. Она превращает эмоционально окрашенный процесс делегирования в прозрачный и контролируемый рабочий процесс.

Целевая аудитория:

Жители города: Основные пользователи, нуждающиеся в простом и оперативном инструменте для сообщения о проблемах (ямы на дорогах, несанкционированные свалки, неработающее освещение). Для них важен мобильный доступ, возможность отслеживать статус своего обращения и влиять на приоритеты через голосование.

Модераторы (представители городских служб): Пользователи, ответственные за обработку и распределение поступающих обращений. Нуждаются в централизованном инструменте для систематизации заявок, контроля их исполнения и предоставления обратной связи жителям. Для них предназначен веб-портал с расширенным функционалом фильтрации и управления.

Внешние аналитики, журналисты и активисты: Потребители обезличенных данных, предоставляемых через публичный API. Нуждаются в доступе к статистике для проведения независимого анализа эффективности работы служб, подготовки медиаматериалов и исследований в области урбанистики.

Обзор аналогов

## Аналог №1 – «Госуслуги. Решаем вместе»

«Госуслуги. Решаем вместе» – это федеральная платформа обратной связи (ПОС), разработанная Минцифры РФ. Она широко применяется гражданами для направления официальных обращений в государственные органы по широкому спектру проблем и интегрирована в основную экосистему «Госуслуг».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2 – Интерфейс «Госуслуги. Решаем вместе»

Преимущества:

* гарантированный официальный статус каждого обращения;
* широкий охват по всей территории РФ и интеграция с «Госуслугами»;
* обращение напрямую поступает в ответственное ведомство.

Недостатки:

* гарантированный официальный статус каждого обращения;
* широкий охват по всей территории РФ и интеграция с «Госуслугами»;
* обращение напрямую поступает в ответственное ведомство.

## Аналог №2 – «SeeClickFix»

SeeClickFix – это коммерческий международный сервис, созданный для взаимодействия жителей и городских властей. Система позволяет публично отмечать проблемы на карте, комментировать их и голосовать, создавая активное сообщество.

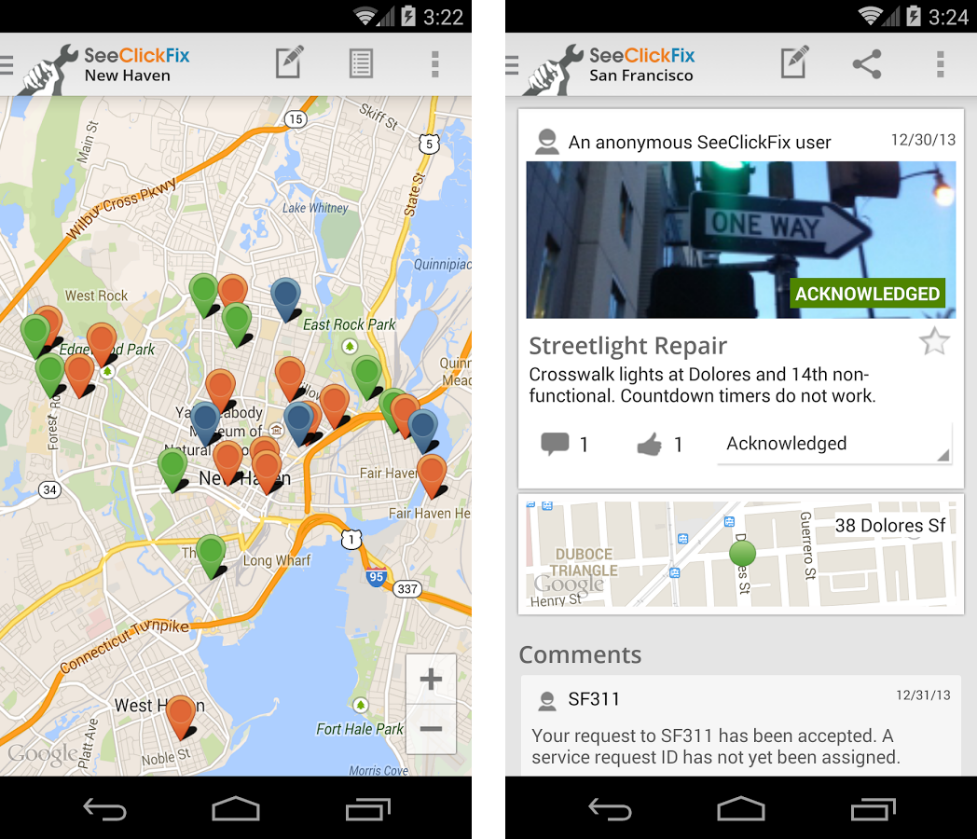


Рисунок 3 – Интерфейс программы «SeeClickFix»

Преимущества:

* современный и интуитивно понятный интерфейс;
* высокая прозрачность: все заявки публичны и доступны на карте;
* развитые социальные функции для вовлечения сообщества (голосование, комментарии).

Недостатки:

* сервис не представлен на российском рынке и не интегрирован с органами власти РФ;
* обращения не имеют официального статуса без платной подписки муниципалитета;
* является коммерческим продуктом для городов.

## Аналог №3 – «Наш Санкт-Петербург»

Многие крупные города РФ разрабатывают собственные порталы для сбора обращений, которые глубоко интегрированы с местными службами. Они решают задачи в рамках одного региона и учитывают его специфику.

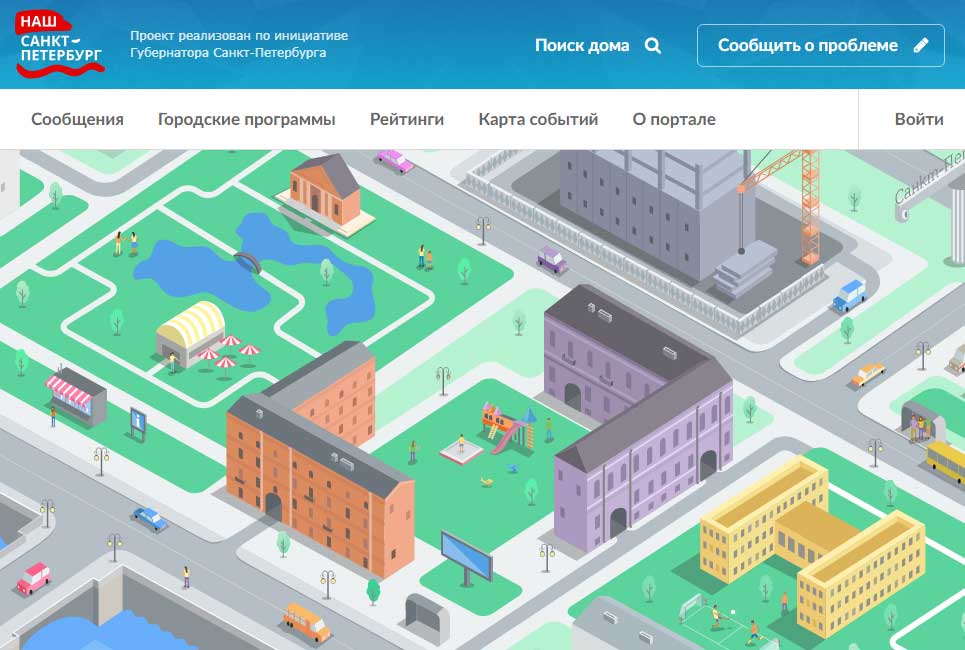


Рисунок 4 – Интерфейс портала «*Наш Санкт-Петербург*»

Преимущества:

* глубокая интеграция с конкретными городскими и районными службами;
* категории проблем точно соответствуют структуре управления городом;
* имеют официальный статус в рамках своего региона.

Недостатки:

* работают только в пределах одного крупного города, не масштабируются на другие населенные пункты;
* как и федеральные платформы, лишены социальных функций;
* данные полностью закрыты, отсутствует Public API.

Вывод

Существующие решения делятся на два типа: официальные, но закрытые и несоциальные государственные порталы, и открытые, социальные, но не представленные в России международные сервисы.

ИС «ПИТТАСИ» спроектирована, чтобы объединить лучшие качества обоих подходов и предложить уникальные возможности:

* прозрачность и социальное взаимодействие: в отличие от государственных аналогов, «ПИТТАСИ» делает все проблемы публичными, а функции голосования и комментирования позволяют формировать общественный запрос и определять реальные приоритеты для жителей, а не только для чиновников;
* универсальность и масштабируемость: в отличие от узкоспециализированных региональных порталов, «ПИТТАСИ» является универсальным решением, которое может быть развернуто в любом городе или населенном пункте России без необходимости дорогостоящей адаптации;
* ключевое преимущество – открытость данных (Public API): это главная особенность, которой нет ни у одного из рассмотренных российских аналогов. Предоставление публичного доступа к обезличенным данным о проблемах создает платформу для независимого анализа эффективности работы служб, для исследований урбанистов и для создания медиапроектов.

Таким образом, разработка ИС «ПИТТАСИ» является актуальной и целесообразной, так как она предлагает уникальное на российском рынке сочетание социальной механики, универсальности и открытости данных, направленное на создание реального инструмента общественного контроля.

# Требования к результатам разработки

## Требования к функциональным характеристикам

Система должна обеспечивать следующий функционал:

* управление пользователями и ролями: регистрация и аутентификация пользователей (Житель города, Модератор), и разделение прав доступа на основе ролей;
* управление обращениями: создание обращений с указанием геолокации, категории проблемы и текстового описания с прикреплением фотоматериалов; отображение всех публичных обращений на интерактивной карте; просмотр детальной информации по каждому обращению;
* социальное взаимодействие и рейтинг: возможность для пользователей голосовать за актуальность проблем, формируя общественный рейтинг, и оставлять публичные комментарии под обращениями;
* модерация и обратная связь: предоставление функционала для Модераторов по изменению статуса обращения ("Новое", "В работе", "Решено"), а также возможность добавлять официальные комментарии и прикреплять фотоотчеты о проделанной работе;
* уведомления: отправка автоматических push-уведомлений автору обращения при изменении его статуса или добавлении официального комментария;
* публичный API: предоставление публичного, документированного API для получения обезличенных данных (статистика по категориям проблем, их статусам, геолокации).

## Требования к показателям назначения

Показатели назначения:

* точность и целостность данных: обеспечение точного определения и хранения геолокационных данных, корректного отображения статусов и сохранение всей прикрепленной пользователями информации;
* простой и интуитивно понятный интерфейс: интерфейс мобильного приложения и веб-портала должен быть доступен для пользователей с любым уровнем технической подготовки;
* отзывчивость и производительность: обеспечение быстрой загрузки карты и меток, а также минимальной задержки при взаимодействии пользователя с интерфейсом.

## Требования к технологическому стеку

В процессе разработки сайта должны использоваться следующие технологии:

* Backend:
* Фреймворк: NestJS (Node.js, TypeScript);
* API: GraphQL;
* ORM: Prisma;
* База данных: PostgreSQL;
* Аутентификация: JWT;
* Мобильное приложение:
* Язык программирования: Dart;
* Фреймворк: Flutter;
* Картографический сервис: Yandex MapKit.

## Требования к пользовательскому интерфейсу

Интерфейс мобильного приложения должен быть интуитивно понятным и обеспечивать легкий доступ ко всему ключевому функционалу. Содержание и структура основных экранов представлены в виде вайрфреймов (см. рисунки 1-6).

Вайрфрейм экрана авторизации представлен на рисунке 1.

Данный экран является точкой входа для зарегистрированных пользователей. Он содержит поля для ввода логина/Email и пароля. При нажатии на кнопку «Войти» система производит аутентификацию пользователя. В случае успеха пользователь перенаправляется на главный экран системы с картой проблем (см. рисунок 3).

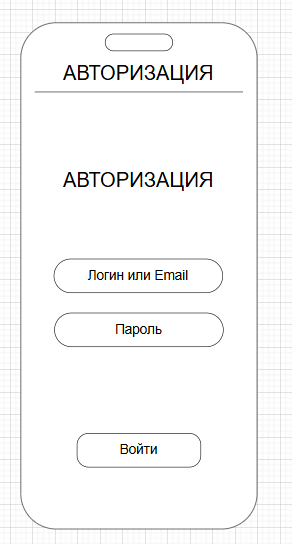


Рисунок 1 – вайрфрейм экрана авторизации

Вайрфрейм экрана регистрации представлен на рисунке 2.

Этот экран предназначен для создания новой учетной записи. Пользователю необходимо заполнить поля «Логин», «Email», «Номер телефона» и задать пароль. Нажатие на кнопку «Зарегистрироваться» инициирует процесс создания аккаунта. После успешной регистрации пользователь автоматически авторизуется и переходит на главный экран (см. рисунок 3).

Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2 – вайрфрейм экрана регистрации

Вайрфрейм главного экрана с картой проблем представлен на рисунке 3.

Это основной экран приложения. На нем отображаются:

* интерактивная карта: на которой геолокационными метками показаны существующие проблемы;
* поисковая строка: позволяет находить проблемы или адреса на карте;
* плавающая кнопка «+»: является ключевым элементом для создания обращения. При нажатии на нее пользователь переходит на экран создания/деталей проблемы (см. рисунок 4);
* нижняя панель навигации: обеспечивает основной переход между разделами. Иконка «Карта» выделена как активная. Нажатие на иконку «Профиль» переходит на экран профиля пользователя (см. рисунок 6).

Изображение выглядит как снимок экрана, диаграмма, текст, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 3 – вайрфрейм главного экрана с картой проблем

Вайрфрейм экрана деталей проблемы и интерфейса ввода комментария представлены на рисунках 4 и 5.

Экран, показанный на рисунке 4, служит двум целям: он используется как форма для создания нового обращения и как страница для просмотра детальной информации о существующей проблеме. На нем отображаются фото, категория, статус и блок с комментариями.

Кнопка «Отправка комментария» инициирует пользовательский ввод. При ее нажатии открывается интерфейс, представленный на рисунке 5, который содержит текстовое поле и кнопку «Отправить» для добавления нового комментария.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 4 – вайрфрейм экрана деталей проблемы

Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 5 – вайрфрейм интерфейса ввода комментария

Вайрфрейм экрана профиля пользователя представлен на рисунке 6.

На данном экране отображается персональная информация о пользователе и его активность в системе. Он включает:

* блок с именем и логином/Email пользователя;
* список «Мои обращения»: динамически отображает все обращения, созданные данным пользователем, с указанием их текущего статуса;
* кнопка «Выйти из аккаунта»: завершает сеанс пользователя и возвращает его на экран авторизации (см. рисунок 1);
* нижняя панель навигации: иконка «Профиль» выделена как активная. Нажатие на иконку «Карта» вернет пользователя на главный экран (см. рисунок 3).

Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 6 – вайрфрейм экрана профиля пользователя

## Требования к видам обеспечения

Данный раздел содержит требования к видам обеспечения разработки.

### Требования к математическому обеспечению

Математическое обеспечение системы должно использовать стандартные алгоритмы для работы с геопространственными данными. Для реализации этих функций на серверной стороне должна использоваться СУБД PostgreSQL с расширением PostGIS.

Основные используемые операции:

* хранение координат (широты и долготы) в специализированных типах данных;
* пространственное индексирование для обеспечения высокой скорости выполнения гео-запросов;
* выполнение запросов для выборки данных в определенной географической области (например, в видимой части карты на экране пользователя).

### Требования к информационному обеспечению

Данный раздел содержит требования к информационному обеспечению.

#### Требования к форматам хранения данных

Информационная база системы должна быть реализована в виде реляционной базы данных под управлением СУБД PostgreSQL. Обмен данными между мобильным приложением и сервером должен осуществляться в формате JSON.

#### Требования к лингвистическому обеспечению

Все пользовательские интерфейсы информационной системы, включая мобильное приложение, должны быть представлены на русском языке.

### Требования к метрологическому обеспечению

Требования к метрологическому обеспечению не предъявляются.

### Требования к техническому обеспечению

Требования для технического обеспечения:

* iOS: 13.0 и новее (iPhone 6s+)
* Android: 8.0 (API 26) и новее
* Оперативная память: 2 ГБ минимум
* Свободное место: 200 МБ для установки
* Минимальная скорость сети: 1 Мбит/с
* Рекомендуемая скорость сети: 3 Мбит/с
* Типы подключения: Wi-Fi, 4G/LTE, 5G
* Минимальное разрешение экрана: 320×480 точек
* Плотность пикселей: от 160 dpi (mdpi)
* Тип ввода: сенсорный экран (multi-touch)

Требуемые разрешения на устройстве:

Обязательные:

* Доступ в Интернет
* Доступ к геолокации (GPS)
* Получение Push-уведомлений

Опциональные:

* Доступ к камере (для создания фотографий проблемы)
* Доступ к галерее/файлам (для прикрепления существующих изображений)

## Требования к надежности

Для обеспечения надежного функционирования программы необходимо принять следующие организационно-технические меры:

* обеспечить бесперебойное питание технических средств.
* осуществлять контроль входных данных для предотвращения ошибок и некорректного поведения программы.
* следовать рекомендациям Министерства труда и социального развития РФ по сервисному обслуживанию ПЭВМ и программного обеспечения.
* регулярно проверять программное обеспечение на наличие компьютерных вирусов в соответствии с требованиями ГОСТ Р ИСО/МЭК 15408 "Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий".

## Требования к безопасности

Основные требования:

Реализуемые решения должны соответствовать нормам электро- и пожаробезопасности в соответствии с требованиями законодательства РФ.

В части требований к информационной безопасности, результат настоящей разработки не должен классифицироваться как информационная система обработки персональных данных.

## Требования к патентной чистоте

Правовая чистота: приложение должно быть разработано с учетом всех правовых норм и не должно нарушать права интеллектуальной собственности. Патентная проверка: приложение должно быть проверено на наличие нарушений патентных прав.

ГК РФ Статья 1354. Патент на изобретение, полезную модель или промышленный образец. Патент на изобретение, полезную модель или промышленный образец удостоверяет приоритет изобретения, полезной модели или промышленного образца, авторство и исключительное право на изобретение, полезную модель или промышленный образец.

## Требования к перспективам развития

Потенциальные функциональные расширения: разработка аналитического модуля для модераторов: внедрение дашбордов с визуализацией данных, включая тепловые карты проблемных зон города, статистику по категориям и средней скорости решения проблем, что позволит городским службам принимать более обоснованные управленческие решения.

Возможные технологические улучшения:

* создание полнофункциональной веб-версии для граждан: разработка веб-портала, дублирующего функционал мобильного приложения, для обеспечения доступности системы с настольных компьютеров;
* реализация полноценного офлайн-режима: предоставление пользователям возможности создавать черновик обращения при отсутствии интернет-соединения с последующей автоматической отправкой при восстановлении связи.

# Состав и содержание работ

В рамках разработки программного продукта в соответствии с настоящим документом необходимо выполнить перечень работ, представленный в таблице 1.

Таблица 1 - перечень работ

| № этапа | Наименование этапа | Состав работ |
| --- | --- | --- |
| 1 | Анализ предметной области | Анализ существующих методов взаимодействия граждан и городских служб, выявление проблем (непрозрачность, низкая скорость реакции) и формулировка требований к системе. |
| 2 | Проектирование системы | Проектирование архитектуры информационной системы, разработка ER-схемы базы данных (PostgreSQL + PostGIS) и создание вайрфреймов пользовательского интерфейса для мобильного приложения. |
| 3 | Реализация системы | Разработка серверной части (Backend на NestJS с GraphQL API) и кроссплатформенного мобильного приложения на Flutter. |
| 4 | Разработка документации | Создание комплекта технической документации (техническое задание, пояснительная записка) и руководства пользователя для мобильного приложения. |
| 5 | Тестирование и сдача продукта | Проведение функционального тестирования, исправление выявленных ошибок, подготовка продукта к приемо-сдаточным испытаниям. |

# Порядок разработки

В рамках разработки программного продукта в соответствии с настоящим документом разработку необходимо проводить в следующем порядке:

Анализ и проектирование

анализ функциональных требований и пользовательских сценариев для ролей «Житель города» и «Модератор»;

проектирование архитектуры системы и GraphQL API;

создание вайрфреймов пользовательского интерфейса для мобильного приложения.

разработка схемы базы данных (PostgreSQL с PostGIS).

Разработка бэкенда

реализация серверной логики на NestJS и GraphQL API;

разработка модуля аутентификации, управления пользователями и ролями;

разработка модуля управления обращениями (CRUD-операции, изменение статусов);

реализация логики голосования и формирования общественного рейтинга проблем;

создание публичного API для предоставления обезличенных данных.

Разработка фронтенда (Мобильное приложение)

создание кроссплатформенного мобильного приложения на Flutter;

реализация экранов для пользовательской роли «Житель города» (карта, создание/просмотр обращения, профиль);

интеграция картографического сервиса (Yandex MapKit) для отображения меток и взаимодействия с картой;

интеграция мобильного приложения с GraphQL API бэкенда;

настройка навигации и состояния приложения.

Тестирование и запуск

проведение функционального и пользовательского тестирования;

исправление выявленных ошибок и оптимизация производительности приложения.

# Требования к документированию

Состав программной документации должен включать в себя: техническое задание, которое должно содержать требования к разработке проекта, его цель, задачи, информацию об исполнителях и заказчиках, сроки и назначении, включает в себя описание предметной области проекта, основные определения и требования к результатам работы, порядку их сдачи и приёмки;

Все вышеперечисленные документы должны быть написаны с учетом требований:

* ГОСТ 34.602-2020
* СТП ВятГУ 101-2004