

Glossary (Глоссарий)

1. Introduction (Введение)

[Данный раздел содержит в себе обзор на весь документ. Данный документ используется для пояснения терминов, которые встречаются в других документах и с которыми читатель может быть не знаком.]

1.1 Purpose (Назначение)

[Укажите назначение данного документа.]

Данный документ содержит в себе определение всех терминов, используемых в документах.

1.2 Scope (Область применения)

[Краткое описание области применения данного документа, к какому(им) проекту(ам) он относится, кем будет использоваться и т.д.]

Документ относится к проекту информационной системы “Бросайте курить!” и относится к аналитикам, product менеджерам, заказчику, разработчикам и тестировщикам с целью унификации терминологии и предотвращения неоднозначности в проектной документации.

1.3 References (Ссылки)

[Перечислите списком названия документов, на которые ссылается в данном, укажите их источники.]

1. Книга “Корпорация “Бросайте курить”, автор Стивен Кинг.
2. Vision.
3. Software Requirements Specification (SRS).
4. Use Case template (UC).
5. Risk List (RL).
6. Business Case (BC).
7. SDP (Software Development Plan).

1.4 Overview (Обзор документа)

[Приведите краткое описание остальных разделов документа.]

В разделе 2 представлены термины, используемые в документах проекта, и их определения.

2. Definitions (Определения)

[В данном разделе описываются термины в алфавитном порядке.]

2.1 <Термин>

[Определение термина в таком объеме, чтобы читатель документа понял, какую именно сущность он описывает.]

1. Диаграмма Ганта - это инструмент для планирования и управления проектами. Это график, на горизонтальной оси которого отмечают действия, на вертикальной оси — продолжительность их выполнения.
2. Лендинг - односторонний сайт, специально разработанный для маркетинговых или рекламных целей.
3. ПО - программное обеспечение, набор программ на компьютере, смартфоне или планшете, который нужен для решения определённых задач.
4. Слежка - процесс наблюдения за клиентом, фиксация случаев рецидива и написание отчетов.
5. ФН - наказание, которое осуществляется при помощи физического воздействия (физическое насилие).
6. ЭВМ - Электронная вычислительная машина, комплекс технических, аппаратных и программных средств, предназначенных для автоматической обработки информации, вычислений, автоматического управления.
7. ЭН - наказание, при помощи воздействия электрического тока.
8. Backend - серверная часть системы, отвечающая за обработку данных, бизнес-логику и взаимодействие с базой данных.
9. CRM-система - программное обеспечение, которое позволяет бизнесу работать с клиентами, собирать заявки, наблюдать за действиями сотрудников и автоматизировать рутину.
10. Frontend - клиентская часть системы, которая отвечает за пользовательский интерфейс и взаимодействие с пользователем.
11. IoT устройство - технические устройства (например, датчики или камеры), подключенные к интернету, которые собирают данные о клиентах (пульс, уровень никотина, запах дыма и др.).
12. IP-адрес — это уникальный адрес, который присваивается устройствам при подключении к интернету или локальной сети.
13. OpenProject - программное обеспечение для управления задачами и проектами.
14. Rocket.Chat - корпоративный мессенджер для общения и координации работы сотрудников.
15. Task-tracker - инструмент для управления задачами, который помогает организовать и отслеживать выполнение различных задач и проектов.
16. Uptime - это процент от общего времени, когда система работает без сбоев.

2.2 <Группа терминов>

[Иногда бывает полезно организовать определения в группы для улучшения читаемости документа. Кратко опишите область данной группы. Термины внутри неё должны располагаться в алфавитном порядке.]

2.2.1 <Термин в группе>

[*Определение термина в таком объеме, чтобы читатель понял, какую именно сущность он описывает.*]

1. Серверная часть

Данная группа объединяет и описывает компоненты, отвечающие за обработку данных, бизнес-логику и взаимодействие с базами данных.

1.1 Гексагональная архитектура - способ проектирования приложения, который используется при разработке программного обеспечения с целью отделить основную логику приложения от сервисов, которые оно использует.

1.2 Проксирование - процесс перенаправления запросов между клиентом и сервером через промежуточный сервер (прокси).

1.3 Checkstyle - инструмент для проверки соответствия кода стандартам оформления.

1.4 Grafana - это платформа для визуализации и анализа данных, которая работает с Prometheus.

1.5 gRPC - это современный высокоскоростной способ обмена данными между разными частями системы (например, между серверами или между приложением и сервером).

1.6 Java - язык программирования, используемый для разработки серверной части.

1.7 Nginx - программное обеспечение с открытым исходным кодом для создания веб-серверов. Оно принимает запрос клиента, например браузера, обрабатывает его и возвращает ответ.

1.8 PostgreSQL - база данных для хранения и управления структурированной информацией.

1.9 Prometheus - это система мониторинга и сбора метрик с открытым исходным кодом.

1.10 react-router-dom — это библиотека для навигации в React-приложениях.

1.11 Spring Boot - фреймворк для быстрой разработки приложений на Java.

2. Клиентская часть

Данная группа объединяет и описывает компоненты, отвечающие за пользовательский интерфейс и взаимодействие с пользователем.

2.1 Axios - библиотека JavaScript для выполнения HTTP-запросов.

2.2 CSS - язык стилей для оформления веб-страниц, отвечает за визуальное представление: цвета, шрифты, отступы.

2.3 ESLint - это инструмент для проверки JavaScript-кода на ошибки и соответствие стилю.

2.4 HTML - язык гипертекстовой разметки, который используется для создания и структурирования веб-страниц.

2.5 JavaScript - язык программирования, используемый для разработки клиентской части.

2.6 Prettier - это инструмент для автоматического форматирования кода.

2.7 React - JavaScript-библиотека для создания пользовательских интерфейсов.

3. Технологические стандарты

Данная группа объединяет и описывает ключевые технологии и протоколы, обеспечивающие работу системы и взаимодействие между компонентами.

3.1 Контейнеризация - технология для развертывания приложений в изолированных контейнерах, что упрощает переносимость и масштабирование.

3.2 CI/CD - способ разработки, в котором тестирование и развертывание кода происходит автоматически.

3.2.1 CI - автоматически проверяет новый код на ошибки при каждом изменении.

3.2.2 CD - автоматически выпускает проверенный код в работу.

3.3 Docker - технология контейнеризации для развертывания приложений.

3.4 HTTP/HTTPS - стандартные протоколы для взаимодействия между клиентской и серверной частями системы. HTTPS обеспечивает дополнительное шифрование данных для защиты конфиденциальной информации.

3.5 MQTT - протокол передачи данных для IoT-устройств, обеспечивающий обмен сообщениями между устройствами и сервером.

3.6 RESTful API - стандартизованный способ взаимодействия компонентов системы через HTTP-запросы (GET, POST, PUT, DELETE) для обеспечения совместимости между клиентом и сервисом.