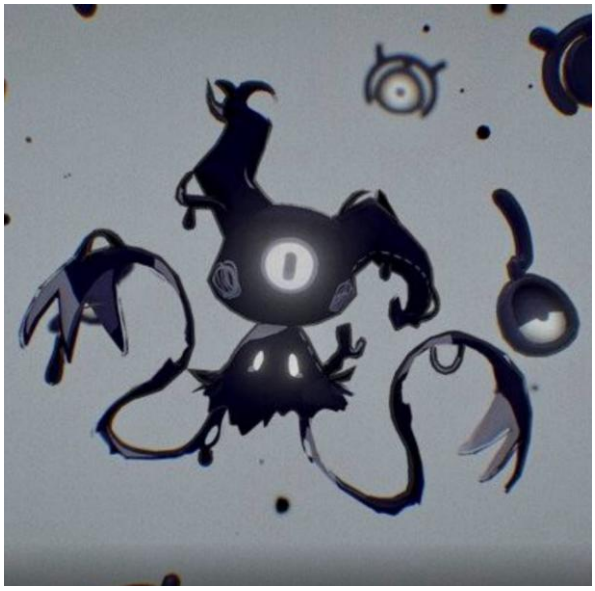
Команда №6

Zawarkich



Спецификация требований к программному обеспечению

Документ

СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ

1.1 Цель

Данный документ представляет собой спецификацию требований к программному обеспечению (SRS) для веб-приложения "Расход". Цель этого документа — предоставить подробное описание функциональных и нефункциональных требований к системе. Этот документ предназначен для использования всеми заинтересованными сторонами, включая разработчиков, тестировщиков, менеджеров проектов и конечных пользователей, для обеспечения общего понимания продукта, который будет разработан.

1.2 Область применения

Система "Расход" — веб-приложение, предназначенное для помощи коллективу в обмене информацией о своем местоположении и причинах отсутствия на рабочем или учебном месте. Оно предоставляет два режима доступа: обычный пользователь для сотрудников и суперпользователь для ответственного за группу лица.
Основные возможности включают отметку местоположения, отслеживание присутствия всей группы в режиме онлайн, систему регистрации, а также автоматическую передачу прав суперпользователя при его отсутствии. Для удобства использования взаимодействие с системой реализовано через Telegram-бота.

Приложение "Расход" охватывает несколько функциональных областей. Управление списками обеспечивает постоянный доступ к актуальным таблицам с информацией от участников. Управление пользователями организует связь с участниками для уточнения их местоположения. Отдельный модуль отвечает за аутентификацию пользователей, включая процессы регистрации, входа и выхода из системы. Пользовательский интерфейс представлен интуитивно понятным и адаптивным веб-интерфейсом, дополненным ботом для удобного взаимодействия с системой.

1.3 Определения и аббревиатуры

SRS (Software Requirements Specification) - спецификация требований к ПО.

Задача - отдельный элемент работы, который должен быть выполнен.

Проект - группа связанных задач.

Пользователь - лицо, взаимодействующее с системой.

1.4 Обзор

В этом документе описываются функциональные и нефункциональные требования к приложению "Расход". Раздел 2 предоставляет общее описание продукта, его перспективы, функции и характеристики пользователей, что будет наиболее интересно заказчикам и конечным пользователям. Раздел 3 содержит детальные, технические требования к системе, предназначенные для разработчиков и тестировщиков.

2 ОБЩЕЕ ОПИСАНИЕ

2.1 Перспектива продукта

Продукт "Расход" является веб-приложением с подключенной к нему базой данных. Он не является частью более крупной системы и не взаимодействует напрямую с другими внешними программными продуктами, за исключением стандартного взаимодействия с веб-браузером пользователя. По своей концепции "Расход" схож с такими системами, как Google Docs, Notion, Obsidian однако представляет собой более легковесное и простое решение, ориентированное на более узкую проблему. В связи с более узкой специализацией улучшены инструменты для работы с правами доступа и связью между пользователями для организации.

2.1.1 Программные интерфейсы

- Система управления базами данных (СУБД) - PostgreSQL
- Telegram Bot API

2.1.2 Пользовательские интерфейсы

Система предоставляет адаптивный веб-интерфейс, доступный через современные веб-браузеры, такие как Chrome, Firefox, Safari и Edge. Этот интерфейс отличается интуитивной понятностью и минималистичным дизайном. Он включает панель управления для суперпользователя, которая обеспечивает обзор статусов всех участников группы. Для обычных пользователей доступен личный кабинет, позволяющий отмечать свое местоположение и просматривать статусы коллег. Кроме того, веб-интерфейс содержит необходимые формы для аутентификации, включая вход в систему и регистрацию.

Параллельно система предлагает альтернативный интерфейс через Telegram-бота. Взаимодействие с ним происходит с помощью текстовых команд, таких как "/start" или "/status", а также через интерактивные кнопки в чате. Эти кнопки позволяют пользователям быстро отмечать свое местоположение, выбирая варианты "В офисе", "Удаленно" или "Отсутствую", без необходимости открывать веб-браузер.

Интерфейсы связи

Взаимодействие между веб-клиентом и сервером, а также между сервером приложения и Telegram Bot API будет осуществляться по защищенному протоколу HTTPS. Это необходимо для обеспечения конфиденциальности и целостности всех передаваемых данных.

Для доступа к базе данных и функционирования веб-сервера будет использоваться стандартное сетевое взаимодействие по протоколу TCP/IP.

2.1.4 Ограничения памяти

Приложение будет развернуто на стандартном виртуальном хостинге или VPS с конфигурацией, типичной для небольших веб-приложений (например, от 512 МБ до 1 ГБ оперативной памяти). Архитектура приложения должна быть спроектирована с учетом эффективного использования ресурсов.

2.2 Функции продукта

Продукт "Расход" предоставляет пользователям комплексный набор функций для управления присутствием и местоположением.

Управление пользователями и аутентификация: в системе реализована регистрация новых пользователей и их последующая аутентификация при входе в веб-приложение или при связывании аккаунта с Telegram-ботом. Права доступа разделены между двумя ролями: "Обычный пользователь" и "Суперпользователь".

Отслеживание местоположения и статуса: пользователи могут оперативно отмечать свое текущее местоположение, выбирая такие варианты, как "На месте", "Удаленно" или "Отсутствую", с возможностью указать причину. Это можно сделать как через веб-интерфейс, так и в Telegram-боте. После отметки статус пользователя в системе автоматически обновляется.

Мониторинг группы в реальном времени: суперпользователь получает доступ к единой панели управления, где может просматривать актуальный статус и местоположение всех участников своей группы. Также система предоставляет возможность отображения истории изменений статусов.

Уведомления и взаимодействие через Telegram: система обеспечивает взаимодействие через мессенджер: суперпользователю рассылаются уведомления о критических изменениях статусов участников. Для всех пользователей Telegram-бот представляет собой удобный и быстрый способ обновлять свой статус напрямую из чата.

2.3 Характеристики пользователей

Предполагаемые пользователи системы делятся на две основные категории:

Обычные пользователи (сотрудники/студенты): члены коллектива, которые должны отмечать свое присутствие. Их

техническая экспертиза может варьироваться от начального до среднего уровня. Это обуславливает необходимость предельно простого и интуитивного интерфейса как в веб-версии, так и в Telegram-боте, с минимальным количеством необходимых действий.

Суперпользователи (руководители групп/ преподаватели): Ответственные лица, которым необходим обзор по всей группе. Их основная задача — мониторинг, поэтому интерфейс для них должен предоставлять информацию в наглядном и структурированном виде, без избыточных деталей. Предполагается, что они имеют достаточный опыт использования базовых веб-приложений.

2.4 Ограничения

Продукт должен функционировать в рамках ряда важных ограничений и требований.

В области аппаратного обеспечения приложение нуждается в оптимизации для работы на стандартных серверах с ограниченными вычислительными ресурсами и объемом памяти.

Что касается взаимодействия с другими системами, критически важным является интерфейс с внешним сервисом Telegram Bot API. Функциональность Telegram-бота напрямую зависит от доступности и стабильности этого API, и любые изменения в нем могут повлиять на работу системы.

Ключевое внимание уделяется требованиям безопасности. Система обязана обеспечивать полную конфиденциальность данных о местоположении пользователей. Эти данные должны быть доступны исключительно участникам их группы и суперпользователю. Дополнительно должна быть реализована надежная и безопасная система аутентификации всех пользователей.

2.5 Допущения и зависимости

Разработка и функционирование продукта "Расход" основываются на ряде ключевых допущений и зависимостей.

Основным допущением является стабильность и постоянная доступность платформы Telegram и ее Bot API на протяжении всего жизненного цикла продукта. Также предполагается, что все целевые пользователи будут иметь постоянный доступ к интернету и необходимому клиентскому программному обеспечению — либо веб-браузеру, либо мессенджеру Telegram.

Что касается зависимостей, то функциональность Telegram-бота полностью определяется корректной работой Telegram Bot API и доступностью сети интернет. Работа веб-интерфейса, в свою очередь, напрямую зависит от стабильности и производительности выбранной

серверной платформы и системы управления базами данных PostgreSQL.

2.6 Распределение требований

Требования, которые могут быть отложены до будущих версий:

Версия 2.0: Расширенная аналитика и отчетность по присутствию за выбранные периоды.

Версия 2.0: Поддержка нескольких групп/проектов для одного пользователя.

Версия 1.1: Push-уведомления в веб-интерфейсе.

Версия 1.1: Настройка внешнего вида и категорий статусов под конкретную группу.

3. СПЕЦИФИЧЕСКИЕ ТРЕБОВАНИЯ

Этот раздел содержит детальные требования к программному обеспечению, достаточные для проектирования системы и тестирования ее соответствия этим требованиям.

3.1 Внешние интерфейсы

3.1.1 Интерфейсы пользователя

Веб-интерфейс: Панель управления (Dashboard_Page)

Основной страницей системы является панель управления с наименованием Dashboard_Page. Она служит единым информационным центром как для суперпользователя, так и для обычного пользователя. Ее главная функция — отображение таблицы с актуальными статусами всех пользователей в группе. Данные на странице представлены в форматах HTML и CSS для базового отображения, а динамическое обновление информации осуществляется через JSON. Организация таблицы включает следующие столбцы: "Имя пользователя", "Текущий статус", "Местоположение", "Время последнего обновления" и "Причина". Для суперпользователя добавляется дополнительный столбец "Действия", содержащий кнопку "Напомнить". В случае отсутствия данных для отображения система показывает соответствующее сообщение: "Данные не найдены".

Веб-интерфейс: Настройка статуса (Status_Update_Modal)

Функция смены статуса пользователем реализована через всплывающее окно или отдельный раздел с наименованием Status_Update_Model. Его основное назначение — предоставить пользователю инструмент для оперативного обновления своего статуса. Входные данные включают выбор из предустановленных значений статуса, таких как "На месте", "Удаленно" и "Отсутствую", а также текстовое поле "Причина", которое является обязательным для заполнения при выборе статуса "Отсутствую". Элементы управления представлены в формате выпадающего списка и текстового поля. Для обеспечения целостности данных реализована валидация: поле "Причина" должно содержать от 1 до 255 символов.

Интерфейс Telegram-бота (Telegram_Bot_Interaction)

Взаимодействие с пользователем через мессенджер организовано в рамках интерфейса Telegram_Bot_Interaction. Общение происходит через чат с использованием текстовых команд. Команда /start инициирует начало работы, запуская процесс регистрации или привязки существующего аккаунта. Команда /status отображает текущий статус пользователя и предоставляет интерфейс для его изменения. Формат вывода

информации — это текстовые сообщения, дополненные клавиатурой быстрых ответов (Inline Keyboard). Набор кнопок включает варианты: "На месте", "Болен" и "Командировка". Входными данными для системы служит выбор пользователя, переданный через механизм callback, который активируется при нажатии одной из кнопок.

3.1.2 Программные интерфейсы

Telegram Bot API

Взаимодействие с мессенджером Telegram осуществляется через официальный Bot API, доступный по адресу https://api.telegram.org/bot<t 8553661923:AAHpQSRQPQKJvz0-9tlu_BscJk505uBIoTA>/. Со стороны системы исходят HTTP POST-запросы в формате JSON. Эти запросы направляются для вызова методов sendMessage, предназначенного для отправки текстовых сообщений и интерактивных кнопок, и answerCallbackQuery, который используется для реакции на нажатия пользователем кнопок в чате. Входящие данные поступают в систему через HTTP POST-запросы от Telegram на специально настроенный веб-хук. Эти запросы содержат объекты Update в формате JSON, которые включают поля message для текстовых команд или callback_query для обработки действий с кнопками.

База данных (PostgreSQL)

База данных PostgreSQL выполняет ключевую роль хранения всех персистентных данных приложения. Связь между сервером приложения и базой данных организуется через соответствующий драйвер, например, pg для среды выполнения Node.js. Взаимодействие строится на выполнении SQL-запросов. Для обеспечения безопасности и предотвращения внедрения кода данные передаются в базу данных исключительно в виде параметризованных запросов.

3.2 Функции

3.2.1 Функция 1: Управление аутентификацией и авторизацией

Требование 1.1: Регистрация пользователя

Система должна предоставлять REST API эндпоинт для регистрации новых пользователей. Входными данными для этого процесса являются email, пароль и имя пользователя. Обязательной является валидация входных данных: адрес электронной почты должен быть уникальным в системе и соответствовать стандартному формату email, а пароль должен содержать не менее 8 символов. Последовательность операций при успешной валидации включает хеширование пароля и сохранение записи о пользователе в базе данных

с присвоением роли "Обычный пользователь" по умолчанию.

Требование 1.2: Аутентификация пользователя

Система должна предоставлять REST API эндпоинт для входа в систему. Процесс аутентификации требует ввода логина и пароля. При успешной проверке учетных данных система возвращает авторизационный токен или устанавливает аутентификационную cookie.

Требование 1.3: Проверка прав доступа

Система должна осуществлять проверку прав доступа, определяемых ролью пользователя, при каждой попытке доступа к защищенным эндпоинтам. Эта проверка особенно критична для таких функциональных модулей, как дашборд суперпользователя, доступ к которому должен быть ограничен исключительно пользователями с соответствующими привилегиями.

3.2.2 Функция 2: Обработка обновления статуса

Требование 2.1: Обновление статуса пользователя

Система должна предоставлять REST API эндпоинт для обновления текущего статуса пользователя. Входными данными для этого эндпоинта являются идентификатор пользователя, новый статус и причина изменения, которая является опциональной для одних статусов, но обязательной для других, например, при отсутствии. Обязательной является валидация входных данных: значение статуса должно строго соответствовать одному из предопределенных в системе значений. Уведомление может быть отправлено через Telegram при условии, что эта функция подключена и настроена.

Требование 2.2: Обработка действий из Telegram

Система должна корректно обрабатывать callback-запросы, поступающие от Telegram Bot API, когда пользователь нажимает на кнопку смены статуса в чате. Входными данными для этого процесса является объект callback_query, содержащий информацию о выбранном статусе и идентификатор пользователя в Telegram. Последовательность операций включает сопоставление полученного Telegram ID с внутренним идентификатором пользователя в системе, вызов стандартной логики обновления статуса и отправку в чат подтверждающего сообщения о успешном изменении статуса и передаче прав суперпользователя.

3.2.3 Функция 3: Автоматическая передача прав суперпользователя

Требование 3.1: Планирование проверки статуса

Система должна ежедневно в 09:00 по серверному времени выполнять автоматическую проверку статуса текущего

суперпользователя.

Требование 3.2: Логика передачи прав

Если в момент проверки статус суперпользователя имеет значение "Отсутствую", и данное состояние сохраняется непрерывно в течение более чем 24 часов, система должна инициировать процедуру передачи прав. Для этого осуществляется поиск самого старшего пользователя в группе, определяемого по самой ранней дате регистрации, который в данный момент имеет активный статус "На месте" или "Удаленно".

Требование 3.3: Назначение прав и уведомление

После идентификации нового суперпользователя система должна автоматически назначить ему соответствующую роль в системе. О данном изменении необходимо уведомить пользователя через все доступные каналы коммуникации, включая сообщение в Telegram и уведомление в веб-интерфейсе.

Требование 3.4: Восстановление первоначальных прав

Система должна обеспечивать автоматическое возвращение прав суперпользователя первоначальному владельцу в момент, когда он изменяет свой статус обратно на активный, то есть на "На месте" или "Удаленно". Процедура восстановления прав должна выполняться немедленно после обновления статуса.

3.2.4 Функция 4: Предоставление данных для дашборда

Система должна предоставлять REST API эндпоинт, предназначенный для получения списка всех пользователей, входящих в группу текущего авторизованного пользователя. Выходные данные эндпоинта должны представлять собой JSON-массив, где каждый элемент является объектом, содержащим следующие данные о пользователе: его уникальный идентификатор, имя, текущий статус, местоположение, временную метку последнего обновления статуса и причину отсутствия, если она указана.

В рамках обработки ошибок система должна проверять права доступа к данному эндпоинту. При отсутствии валидной авторизации или недостаточных прав доступа запрос должен быть отклонен с возвратом соответствующего HTTP-кода состояния 403 (Forbidden) с перенаправлением на главную страницу для корректной обработки запроса.

3.3 Требования к производительности

Подавляющее большинство API-запросов, за исключением эндпоинта загрузки дашборда, должны выполняться менее чем за 500 миллисекунд. Данное требование должно соблюдаться при нагрузке, достигающей 50 одновременных пользователей. Эндпоинт загрузки

дашборда имеет отдельный критерий производительности — он должен возвращать данные менее чем за 2 секунды при условии, что количество пользователей в группе не превышает 100 человек.

Требование к параллельной обработке

Система должна быть рассчитана на поддержку до 10 одновременных подключений пользователей через веб-интерфейс. Параллельно с этим система должна обрабатывать до 100 одновременных сессий взаимодействия, осуществляемых через Telegram-бота.

Требование к скорости уведомлений

Обработка событий и последующая рассылка уведомлений через мессенджер Telegram должны происходить в течение 10 секунд с момента возникновения инициирующего события в системе.

3.4 Логические требования к данным

- **Сущность** User**:**
  - id (UUID, Primary Key)
  - email (VARCHAR(255), Unique, Not Null)
  - password_hash (VARCHAR(255), Not Null)
  - name (VARCHAR(100), Not Null)
  - telegram_chat_id (BIGINT, Unique, Nullable)
  - role (ENUM('user', 'superuser'), Default: 'user')
  - created_at (TIMESTAMP)
- **Сущность** StatusLog**:**
  - id (UUID, Primary Key)
  - user_id (UUID, Foreign Key to User.id)
  - status (ENUM('in_office', 'remote', 'absent'), Not Null)
  - reason (TEXT, Nullable)
  - created_at (TIMESTAMP)
  -

3.5 Атрибуты программной системы

3.5.1 Надежность (Reliability)

Среднее время наработки на отказ системы должно составлять не менее 720 часов, что эквивалентно одному месяцу непрерывной работы в рабочей среде. Этот показатель характеризует общую стабильность и надежность платформы.

Требование к обработке внешних сбоев

Система должна быть спроектирована для корректной обработки ошибок связи с внешними сервисами, в частности с Telegram API. Сбои в работе внешнего API не должны приводить к остановке или нарушению основного функционала веб-приложения. Все возникающие

ошибки межсистемного взаимодействия должны фиксироваться путем записи в логи для их последующего анализа и устранения причин.

3.5.2 Доступность (Availability)

Система должна обеспечивать доступность 99.5% в течение рабочего дня (с 08:00 до 20:00 по времени сервера). Плановые технические работы, уведомленные за 24 часа, в этот показатель не входят.

3.5.3 Безопасность (Security)

Все пароли пользователей должны храниться в защищенном виде с использованием алгоритма хеширования. Для этого должен применяться стойкий криптографический алгоритм, такой как bcrypt, который включает добавление "соли" для противодействия атакам перебора.

Требование к защите передаваемых данных

Все данные, передаваемые между клиентскими приложениями и сервером, должны быть защищены с помощью протокола HTTPS. Минимальная требуемая версия протокола TLS — 1.2, при этом рекомендуется использование более современных версий для обеспечения повышенной безопасности.

Требование к разграничению доступа к данным

Доступ к конфиденциальной информации, такой как данные о статусе и местоположении пользователей, должен быть строго ограничен. Пользователь может просматривать данные только в пределах своей группы. Например, пользователь, принадлежащий к группе "А", не должен иметь возможности доступа к каким-либо данным пользователей из группы "Б".

Требование к валидации входных данных

Все без исключения данные, поступающие от пользователей, независимо от канала поступления — будь то веб-формы или сообщения из Telegram-бота — должны проходить обязательную проверку на стороне сервера. Эта проверка должна включать защиту от внедрения вредоносного кода и проверку на соответствие ожидаемым типам и форматам данных.

3.5.4 Сопровождаемость (Maintainability)

Кодовая база должна быть модульной. Логика работы с БД, бизнес-логика и API-роуты должны быть разделены.

Система должна использовать централизованное логирование. Логи должны содержать timestamp, уровень логирования (INFO,

ERROR) и идентификатор запроса для трассировки.

3.5.5 Удобство использования (Usability)

Для выполнения основной операции (смена статуса) в веб-интерфейсе должно требоваться не более 3 кликов. В Telegram-боте - не более 2 действий (открыть чат, нажать кнопку).

Интерфейс должен быть локализован на русский язык.

4. ПРОЦЕСС УПРАВЛЕНИЯ ИЗМЕНЕНИЯМИ

Для обеспечения контроля и отслеживаемости изменений в требованиях к проекту "Расход" устанавливается формализованный процесс управления изменениями.

Идентификация и Регистрация

Любое предложение по изменению, именуемое Запросом на Изменение (Change Request, CR), подлежит обязательной фиксации. Источником такого запроса может выступить любая из сторон: заказчик, команда разработки, тестировщики или менеджер проекта. Все поступающие запросы регистрируются в централизованном Журнале изменений, который может быть реализован в виде таблицы в Confluence, Excel или специализированной системе, такой как Jira. Каждому CR присваивается уникальный идентификатор, а также фиксируется краткое описание, дата подачи, инициатор и исходный приоритет.

Подача Запроса

Все запросы на изменение должны быть поданы в письменной форме. Предпочтительным способом является создание задачи в системе управления проектами с соответствующей пометкой "CR" или отправка формализованного электронного письма на адрес менеджера проекта. Предложения, высказанные устно в ходе личных бесед или телефонных разговоров, не подлежат рассмотрению до момента их официальной письменной фиксации.

Анализ и Оценка

Менеджер проекта формирует комиссию по рассмотрению изменений (Change Control Board, CCB), в состав которой входят представители заказчика, ведущий разработчик и ведущий тестировщик. Данная комиссия проводит всестороннюю оценку каждого запроса, анализируя его потенциальное влияние на сроки реализации проекта, общую трудоемкость и бюджет, архитектурные и дизайнерские решения системы, а также на существующий набор функциональных требований. Результатом работы комиссии является заключение с одной из рекомендаций: принять изменение, отклонить его или отложить реализацию до следующей версии продукта.

Утверждение и Реализация

Заключение, подготовленное комиссией CCB, направляется на утверждение заказчику. После получения утверждения менеджер проекта вносит необходимые корректировки в Документ Требований к Программному Обеспечению и другие связанные документы. Все изменения в SRS должны быть четко отслеживаемы, для чего применяется версионирование документа. Утвержденный запрос на изменение переносится в бэклог продукта или в план очередного

спринта для непосредственной реализации разработчиками.

Информирование

О всех принятых решениях и соответствующих обновлениях в документации оперативно информируются все заинтересованные стороны проекта. Это обеспечивает прозрачность процесса и поддерживает актуальность информации у всех участников.