Специфікація програмного забезпечення

Програмна система для моніторингу та оцінки техніки виконання фізичних вправ

Software Requirements Specification

1.0

13.06.2025

Ткаченко Євген Андрійович

**ІСТОРІЯ ЗМІН**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Опис** | **Автор** | **Коментарі** |
| 13.06.2025 | Створено пункти 1.1 - 1.5 | Ткаченко Євген Андрійович |  |
| 13.06.2025 | Створено пункти 2.1 - 2.5 | Ткаченко Євген Андрійович |  |
| 13.06.2025 | Створено пункти 3.1 - 3.4 | Ткаченко Євген Андрійович |  |
| 13.06.2025 | Створено пункт 3.5 | Ткаченко Євген Андрійович |  |

**ЗАТВЕРДЖЕННЯ ДОКУМЕНТУ**

Наступну специфікацію вимог до програмного забезпечення було прийнято та схвалено:

|  |  |  |  |
| --- | --- | --- | --- |
| **Підпис** | **Друковане ім’я** | **Назва** | **Дата** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**ЗМІСТ**

ІСТОРІЯ ЗМІН

ЗАТВЕРДЖЕННЯ ДОКУМЕНТУ

1. ВСТУП

1.1 Огляд продукту

1.2 Мета

1.3 Межі

1.4 Посилання

1.5 Означення та абревіатури

2. ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

2.2 Функції продукту

2.3 Характеристики користувачів

2.4 Загальні обмеження

2.5 Припущення й залежності

3. КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

3.1.2 Апаратний інтерфейс

3.1.3 Програмний інтерфейс

3.1.4 Комунікаційний протокол

3.1.5 Обмеження пам’яті

3.1.6 Операції

3.1.7 Функції продукту

3.1.8 Припущення й залежності

3.2 Властивості програмного продукту

3.3 Атрибути програмного продукту

3.3.1 Надійність

3.3.2 Доступність

3.3.3 Безпека

3.3.4 Супроводжуваність

3.3.5 Переносимість

3.3.6 Продуктивність

3.4 Вимоги бази даних

3.5 Інші вимоги

1. ВСТУП

1.1 Огляд продукту

Метою даного продукту є створення онлайн-платформи для моніторингу та оцінки техніки виконання підтягувань із підтримкою IoT-сенсорів. Система приймає телеметрію (висоту підйому, кут нахилу тулуба, час між повтореннями), візуалізує ці метрики в особистому кабінеті користувача та автоматично формує рекомендації для покращення техніки.

Цільова аудиторія охоплює:

* спортсменів-аматорів, які тренуються вдома й потребують об’єктивного контролю;
* просунутих атлетів і персональних тренерів, що аналізують прогрес секцій або груп;
* адміністраторів фітнес-клубів, які прагнуть надати клієнтам «розумні» снаряди з телеметрією.

1.2 Мета

Платформа повинна забезпечити:

* збір і зберігання біомеханічних показників кожного повторення;
* автоматичне розпізнавання помилок (недостатня висота, надмірний нахил, нерівномірний темп);
* генерацію персональних підказок у реальному часі;
* централізоване адміністрування користувачів і тренувальних сесій;
* доступність із будь-якого браузера без необхідності складного калібрування.

1.3 Межі

Продукт охоплює:

* ESP32-датчики (акселерометр + ультразвук) → HTTP POST → Spring Boot 17 REST-API;
* PostgreSQL 16 для зберігання користувачів, сесій, метрик, рекомендацій;
* SPA на Vue 3 + TailwindCSS + Pinia + Chart.js;
* Docker Compose + NGINX-reverse-proxy.

За межами проєкту: підтримка інших вправ, мобільний застосунок, інтеграція з медичними діагностичними пристроями.

* 1. Посилання
* ISO/IEC 25010 – Якісні характеристики ПЗ: (<https://www.iso.org/standard/35733.html>);
* Vue.js 3 Docs – (<https://vuejs.org/>);
* Spring Boot Documentation – (<https://spring.io/>);
* PostgreSQL Docs – (<https://www.postgresql.org/docs/>);
* Docker Compose Reference – (https://docs.docker.com/compose/).

1.5 Означення та абревіатури

Означення:

* Метрика – запис одного повторення вправи, що містить висоту стрибка, кут виконання, час у повітрі та оцінку якості;
* Сесія – тренувальна сесія, яка складається з послідовності метрик, зібраних під час одного заняття;
* Рекомендація – автоматично сформований текст із порадами для покращення результатів на основі аналізу метрик.

Абревіатури:

* SPA (Single Page Application) – односторінковий вебзастосунок;
* REST (Representational State Transfer) – архітектурний стиль для API;
* API (Application Programming Interface) – інтерфейс прикладного програмування;
* DB (Database) – база даних;
* IoT (Internet of Things) – інтернет речей, технологія для підключення пристроїв;
* ESP32 – мікроконтролер для IoT-пристроїв;
* UI/UX (User Interface/User Experience) – користувацький інтерфейс та досвід взаємодії.

2. ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

Платформа може масштабуватися до інших силових вправ (віджимання, присідання), інтегруватися з фітнес-браслетами для пульсомоніторингу, а також слугувати освітнім інструментом на уроках фізкультури чи кафедрах спортивної науки.

* 1. Функції продукту
* створення тренувальної сесії;
* прийом потокової телеметрії від ESP32;
* автоматичний інкремент лічильника повторень;
* побудова графіків:  
   – висота підйому;  
   – кут нахилу тулуба;  
   – інтегральний correctnessScore;
* евристичний аналіз і текстові рекомендації;
* інтерактивна історія сесій;
* експорт вибраної сесії у CSV / JSON (для сторонньої аналітики);
* адмін-панель: CRUD користувачів;
* REST-API для сторонніх клієнтів.
  1. Характеристики користувачів

Користувачі поділяються на декілька категорій, кожна з яких має унікальний набір прав:

* Гість — Перегляд лендингу, реєстрація, запуск/завершення сесії, перегляд графіків, рекомендації.
* Адміністратор — CRUD користувачів.

Такий розподіл забезпечує чітку ізоляцію дій і зменшує ризики помилок або порушень.

2.4 Загальні обмеження

При розробці та впровадженні системи слід враховувати низку обмежень:

* перегляд графіків, рекомендації.ESP32 працює лише у мережах 2.4 GHz Wi-Fi;
* необхідна первинна калібрація датчиків (≤ 1 хв);
* клієнтський кеш браузера ≤ 100 МБ;
* затримка від датчика до відображення – ≤ 1000 мс;
* сервіс не надає медичних діагнозів, рекомендації мають інформаційний характер.
  1. Припущення й залежності

Проєкт має наступні припущення та залежності:

* браузерна підтримка. Платформа повинна підтримувати останні версії Chrome, Firefox, Edge. Тестування на Safari або старих Android WebView не гарантує повної сумісності. Браузери користувачів підтримують ES2020;
* сервер і БД запускаються у Linux-контейнерах Docker Compose;
* IoT-модуль використовує HTTP POST для надсилання телеметрії;
* масштабування кластера планується через Kubernetes.

3. КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

У цьому розділі визначаються, як система буде взаємодіяти з іншими системами, користувачами та пристроями. Деталізується, яким чином продукт обмінюватиметься інформацією з іншими системами чи пристроями.

3.1.1 Інтерфейс користувача

Інтерфейс вебдодатка має бути інтуїтивно зрозумілим і адаптивним до різних розмірів екранів (мобільні пристрої, планшети, настільні ПК). Передбачено підтримку світлої та темної теми. Основні елементи: навігаційне меню, панель поточної сесії, лічильник повторень, інтерактивні графіки висоти та кута нахилу, таблиця історичних сесій, модуль текстових рекомендацій і адмін-панель керування користувачами. Спортсмен може запускати й зупиняти тренувальну сесію, переглядати результати й експортувати дані; тренер – переглядати статистику декількох спортсменів; адміністратор – виконувати повний CRUD.

3.1.2 Апаратний інтерфейс

Вебчастина працює у сучасних браузерах (Chrome, Firefox, Edge, Safari). Для коректної роботи рекомендуються пристрої з ≥4 ГБ RAM, двоядерним процесором ≥2 ГГц та стабільним інтернет-з’єднанням від 5 Мбіт/с. IoT-модуль базується на ESP32 (2,4 GHz Wi-Fi): акселерометр/гіроскоп для кута нахилу й ультразвуковий сенсор для висоти підйому.

3.1.3 Програмний інтерфейс

Додаток на стороні сервера реалізовано у вигляді RESTful API (Java Spring Boot). Передбачено енд-поїнти для:

* реєстрації / входу;
* створення сесій;
* прийому телеметрії (HTTP POST JSON < 1 с затримки);
* отримання статистики й рекомендацій;
* адмін-операцій CRUD.

Для зберігання використовується PostgreSQL. Фронтенд (Vue 3) взаємодіє з API через HTTPS; IoT-модуль надсилає метрики тим самим REST-інтерфейсом.

3.1.4 Комунікаційний протокол

Передача даних здійснюється виключно через HTTPS із підтримкою TLS 1.3. ESP32 надсилає дані HTTP POST-запитом. Передбачена підтримка CORS для безпечної взаємодії між фронтендом та API.

3.1.5 Обмеження пам’яті

Вебклієнт не повинен споживати понад 200 МБ ОЗП при стандартному навантаженні. На ESP32 кеш телеметрії обмежено 64 КБ. База даних оптимізується за допомогою індексів і партиціювання; старі сирі метрики можуть архівуватися.

* + 1. Операції

Користувачі можуть:

* створювати / завершувати тренувальні сесії;
* відправляти телеметрію (ESP32 → API);
* переглядати графіки, статистику та текстові поради;
* переглядати історію тренувань.

Адміністратор виконує CRUD для користувачів, вправ і сесій.

3.1.7 Функції продукту

Система забезпечує:

* Реєстрація, вхід;
* Запуск тренувальної сесії й прийом потокових метрик у реальному часі;
* Візуалізація: графіки висоти та кута, лічильник повторень, середні значення;
* Евристичний розрахунок correctnessScore, формування текстових рекомендацій;
* Адмін-панель CRUD користувачів.

3.1.8 Припущення й залежності

Система працює в середовищі Linux (Docker). Для розгортання використовується Docker Compose. Очікується, що користувачі мають стабільне інтернет-зʼєднання. Підтримка браузерів — останні версії Chrome, Firefox. Система сумісна з будь-якою SQL-сумісною СУБД, але орієнтована на PostgreSQL. За потреби горизонтально масштабується в Kubernetes. Клієнтські браузери підтримують ES2020. ESP32 вимагає стабільного Wi-Fi 2,4 GHz.

3.2 Властивості програмного продукту

Програмне забезпечення реалізує розгалужену логіку моніторингу та оцінки техніки виконання підтягувань із підтримкою декількох ролей (спортсмен, тренер, адміністратор). Система здатна працювати як у режимі одного користувача, так і в мережевій конфігурації з кількома клієнтами та віддаленим адмініструванням. Завдяки мікросервісному підходу підтримується гнучке масштабування й додавання нових модулів — наприклад аналітики інших вправ, мобільного застосунку або розширеної статистики групових тренувань.

3.3 Атрибути програмного продукту

3.3.1 Надійність

Система побудована з урахуванням обробки всіх виняткових ситуацій, дублюванням вимірювань у кеші ESP32 та резервним копіюванням бази даних через cron-job. Docker Compose перезапускає контейнери у разі збою, забезпечуючи безперервну роботу 24/7.

3.3.2 Доступність

Доступ до вебклієнта можливий з будь-якого сучасного браузера. Контейнеризація та можливе розгортання в Kubernetes дозволяють швидко масштабувати бекенд та балансувати навантаження при пікових сесіях.

3.3.3 Безпека

Передбачено валідацію запитів, обмеження прав доступу, CORS. Дані шифруються при передачі. Фото зберігаються в окремому об’єктному сховищі MinIO. Паролі зберігаються у вигляді bcrypt-хешів.

3.3.4 Супроводжуваність

Код дотримується стандартів Java Code Conventions і Vue Style Guide. CI/CD налаштовано через GitHub Actions — автоматичне тестування, побудова контейнерів і розгортання на staging/production.

3.3.5 Переносимість

Увесь стек (PostgreSQL, Spring Boot, Vue.js, NGINX) запаковано в Docker-контейнери, що забезпечує незалежність від ОС хоста й можливість швидкого перенесення на інший сервер або у хмару.

3.3.6 Продуктивність

Система оптимізована для обробки до 100 одночасних активних тренувальних сесій із затримкою отримання метрик ≤ 1 с. Часті запити кешуються у Redis; для звітів застосовуються матеріалізовані представлення.

3.4 Вимоги бази даних

PostgreSQL зберігає дані про користувачів, вправи, сесії, метрики, рекомендації. Усі ключові поля проіндексовані; для вибірок за історією сесій створені materialized views. Логи подій і телеметрії зберігаються окремо для аналітики. Міграції виконуються Flyway.

3.5 Інші вимоги

Захист персональних даних. Система відповідає вимогам GDPR; користувач може експортувати чи видалити свої дані.

Правові обмеження. Програма не формує медичних діагнозів; рекомендації — лише тренувальні поради.

Енергоефективність. Контейнерний підхід і моніторинг ресурсів дозволяють оптимізувати споживання CPU/RAM; неблокуючі I/O та кешування мінімізують навантаження на інфраструктуру.