Специфікація вимог до програмного забезпечення

Програмна система для розповсюдження та споживання текстового та статичного візуального контенту «Read and Feed»

Версія 1.0

Сивуха Ярослав, Турчин Олександр, Дмитро Горбатенко

ПЗПІ-21-5

11.06.2025

**ЗМІСТ**

[1 ВСТУП 4](#1%25191_%25D0%2592%25D0%25A1%25D0%25A2%2)

[1.1 Огляд продукту 4](#1%25191.1_%25D0%259E%25D0%25B3%25D0%25BB)

[1.2 Мета 4](#1%25191.2_%25D0%259C%25D0%25B5%25D1%2582)

[1.3 Межі 5](#1%25191.3_%25D0%259C%25D0%25B5%25D0%25B6)

[1.4 Посилання 6](#1%25191.4_%25D0%259F%25D0%25BE%25D1%2581)

[1.5 Означення та абревіатури 6](#1%25191.5_%25D0%259E%25D0%25B7%25D0%25BD)

[2 ЗАГАЛЬНИЙ ОПИС 7](#1%25192_%25D0%2597%25D0%2590%25D0%2593%2)

[2.1 Перспективи продукту 7](#1%25192.1_%25D0%259F%25D0%25B5%25D1%2580)

[2.2 Функції продукту 8](#1%25192.2_%25D0%25A4%25D1%2583%25D0%25BD)

[2.3 Характеристики користувачів 9](#1%25192.3_%25D0%25A5%25D0%25B0%25D1%2580)

[2.4 Загальні обмеження 10](#1%25192.4_%25D0%2597%25D0%25B0%25D0%25B3)

[2.5 Припущення й залежності 11](#1%25192.5_%25D0%259F%25D1%2580%25D0%25B8)

[3 КОНКРЕТНІ ВИМОГИ 13](#1%25193_%25D0%259A%25D0%259E%25D0%259D%2)

[3.1 Вимоги до зовнішніх інтерфейсів 13](#1%25193.1_%25D0%2592%25D0%25B8%25D0%25BC)

[3.1.1 Інтерфейс користувача 13](#1%25193.1.1_%25D0%2586%25D0%25BD%25D1%25)

[3.1.2 Апаратний інтерфейс 13](#1%25193.1.2_%25D0%2590%25D0%25BF%25D0%25)

[3.1.3 Програмний інтерфейс 14](#1%25193.1.3_%25D0%259F%25D1%2580%25D0%25)

[3.1.4 Комунікаційний протокол 14](#1%25193.1.4_%25D0%259A%25D0%25BE%25D0%25)

[3.1.5 Обмеження пам’яті 14](#1%25193.1.5_%25D0%259E%25D0%25B1%25D0%25)

[3.1.6 Операції 15](#1%25193.1.6_%25D0%259E%25D0%25BF%25D0%25)

[3.2 Властивості програмного продукту 15](#1%25193.2_%25D0%2592%25D0%25BB%25D0%25B0)

[3.3 Атрибути програмного продукту 16](#1%25193.3_%25D0%2590%25D1%2582%25D1%2580)

[3.3.1 Надійність 16](#1%25193.3.1_%25D0%259D%25D0%25B0%25D0%25)

[3.3.2 Масштабованість 16](#1%25193.3.2_%25D0%259C%25D0%25B0%25D1%25)

[3.3.3 Відкрита до відлагодження 16](#1%25193.3.3_%25D0%2592%25D1%2596%25D0%25)

[3.3.4 Адаптивність 16](#1%25193.3.4_%25D0%2590%25D0%25B4%25D0%25)

[3.3.5 Швидкодія 16](#1%25193.3.5_%25D0%25A8%25D0%25B2%25D0%25)

[3.3.6 Підтримка багатьох браузерів 16](#1%25193.3.6_%25D0%259F%25D1%2596%25D0%25)

[3.3.7 Безпека даних 17](#1%25193.3.7_%25D0%2591%25D0%25B5%25D0%25)

[3.3.8 Робота з одночасним доступом 17](#1%25193.3.8_%25D0%25A0%25D0%25BE%25D0%25)

[3.3.9 Захист від помилок 17](#1%25193.3.9_%25D0%2597%25D0%25B0%25D1%25)

[3.4 Вимоги бази даних 17](#1%25193.4_%25D0%2592%25D0%25B8%25D0%25BC)

**1 ВСТУП**

1.1 Огляд продукту

Цей документ детально описує вимоги до програмної системи «Read and Feed» — інноваційної, багатофункціональної веб-платформи, розробленої для розповсюдження, споживання та ефективного управління текстовим та статичним візуальним контентом. У сучасному цифровому світі, де зручність доступу до інформації та розваг є ключовою, «Read and Feed» прагне стати централізованим рішенням для різноманітних форматів читання, включаючи електронні книги, мангу, комікси та авторські твори. Платформа надає користувачам інтуїтивно зрозумілий інтерфейс для пошуку, фільтрування, читання та додавання нового контенту. Система проектується як комплексна екосистема, що включає як потужну серверну частину (Back-end) для обробки даних, безпеки та управління контентом, так і гнучку клієнтську частину (Front-end) разом з мобільним додатком (Mobile) для забезпечення оптимального користувацького досвіду на різних пристроях. Особливий акцент робиться на підтримці багатомовності, включаючи українську мову, що відрізняє «Read and Feed» від багатьох існуючих аналогів, які часто обмежені у мовній підтримці та різноманітності контенту.

1.2 Мета

Основною метою розробки програмної системи «Read and Feed» є створення передової, масштабованої та безпечної веб-платформи, яка задовольнить зростаючий попит на якісний цифровий контент та спростить процеси його споживання та розповсюдження. Зокрема, система має на меті:

* **сприяти поширенню культури читання:** Шляхом надання легкого та зручного доступу до великої бібліотеки різноманітного контенту, «Read and Feed» прагне відновити інтерес користувачів, особливо молодого покоління, до читання;
* **забезпечити багатомовну підтримку:** Однією з ключових цілей є повноцінна підтримка української мови, що дозволить залучити ширшу аудиторію та зробити контент доступнішим для україномовних користувачів, які часто стикаються з обмеженнями на існуючих платформах;
* **уніфікувати досвід споживання контенту:** Об'єднати різні формати (електронні книги, манга, комікси) в єдиній, злагодженій системі, усуваючи необхідність використання кількох спеціалізованих платформ;
* **підвищити зручність та ефективність:** Забезпечити інтуїтивно зрозумілий інтерфейс, швидкий пошук, ефективні фільтри та персоналізовані рекомендації для максимально комфортного та продуктивного використання платформи;
* **забезпечити надійність та безпеку:** Розробити систему з високим рівнем стійкості до відмов, захисту даних користувачів та контенту, а також відповідності сучасним стандартам безпеки.

1.3 Межі

Цей документ охоплює повну специфікацію вимог до **Програмної системи для розповсюдження та споживання текстового та статичного візуального контенту «Read and Feed»** як цілісного програмного продукту. У межах даної специфікації будуть визначені:

1. **Функціональні вимоги:** Детальний опис всіх можливостей системи, доступних для різних категорій користувачів (неавторизовані користувачі, авторизовані користувачі, адміністратори), включаючи пошук, фільтрування, читання, завантаження та управління контентом, управління профілем, а також функціонал модерації.

**Вимоги до зовнішніх інтерфейсів:** Специфікація користувацького інтерфейсу (UI/UX), апаратних вимог для коректної роботи системи, програмних інтерфейсів (API) для взаємодії між компонентами системи та, за необхідності, з зовнішніми сервісами, а також комунікаційних протоколів.

**Вимоги до даних:** Структура та управління базою даних, включаючи схему даних для зберігання контенту, профілів користувачів, метаданих та інших системних об'єктів.

**Нефункціональні вимоги:** Деталізація вимог щодо продуктивності, масштабованості, надійності, безпеки, зручності використання, підтримуваності та сумісності системи. Це включає вимоги до часу відгуку, здатності обробляти значні обсяги контенту та велику кількість одночасних користувачів.

**Дана специфікація не охоплює наступні аспекти:**

* **деталі розгортання та інфраструктури.** Специфічні вимоги до серверного обладнання, операційних систем, які будуть визначатися на етапі розгортання;
* **маркетингові стратегії та бізнес-планування:** Питання, пов'язані з просуванням продукту на ринку, монетизацією контенту;
* **управління проектом та методології розробки:** Ця специфікація зосереджена на що повинна робити система, а не на як вона буде розроблятися (методології розробки, такі як Agile, Scrum, Kanban, а також управління ризиками, бюджетом та розкладом, є предметом окремих планів проекту);
* **детальний дизайн інтерфейсу користувача та графічні елементи:** Хоча UI/UX вимоги будуть окреслені.

1.4 Посилання

Документ посилається на стандарт ДСТУ 3008:2015 «Документація. Звіти у сфері наукових досліджень та розробок. Структура і правила оформлення». Текст має офіційний стиль, IEEE Std 830-1993 «Рекомендована практика IEEE для специфікацій вимог до програмного забезпечення».

1.5 Означення та абревіатури

UI – User Interface

ПЗ – програмне забезпечення

DTO – Data Transfer Object

CI/CD — Continuous Integration Continuos Development

БД – база даних

**2 ЗАГАЛЬНИЙ ОПИС**

2.1 Перспективи продукту

«Read and Feed» позиціонується як самостійний, повноцінний програмний продукт, що прагне зайняти значну нішу на ринку цифрового контенту, виступаючи як комплексна платформа для розповсюдження та споживання текстових та статичних візуальних творів. Хоча система може функціонувати як незалежний сервіс, її архітектура розроблена таким чином, щоб забезпечити потенційну інтеграцію з майбутніми або існуючими зовнішніми платформами та сервісами, які можуть розширити її функціональні можливості або базу користувачів. Наприклад, у довгостроковій перспективі можливе підключення платіжних систем для монетизації контенту, інтеграція з соціальними мережами для покращення обміну інформацією та спільнотою, або ж API для взаємодії з електронними бібліотеками та освітніми ресурсами.

Основними плюсами системи є:

* **кінцеві користувачі (читачі):** Отримують єдину, зручну та багатомовну платформу для доступу до широкого спектру контенту (книги, манга, комікси) без необхідності перемикатися між різними сервісами і ефективний інструмент для публікації, просування та, за бажанням, монетизації своїх творів, досягаючи ширшої аудиторії та отримуючи прямий зворотний зв'язок;
* **адміністратори платформи:** Отримують потужні засоби для управління контентом, користувачами, модерації та моніторингу активності системи.

Система розробляється з використанням сучасних технологій, що гарантує її актуальність та сумісність з сучасними веб-стандартами та пристроями, забезпечуючи тривалий життєвий цикл продукту.

2.2 Функції продукту

**Неавторизований користувач матиме такі функції:**

* **перегляд контенту:** Можливість переглядати доступний контент (обмежений за кількістю розділів/сторінок або за типом контенту) без реєстрації;
* **пошук та фільтрація контенту:** Доступ до функціоналу пошуку за назвою, автором, жанром, тегами та основними фільтрами;
* **ознайомлення з інформацією про контент:** Перегляд обкладинок, описів, метаданих (жанр, рік видання, автор) та коментарів користувачів;
* **реєстрація та авторизація:** Можливість створення нового облікового запису та вхід до системи.

**Функції які будуть доступні авторизованим користувачам:**

* **персоналізована бібліотека:** Створення та управління власною бібліотекою контенту (додавання до списку бажаного та улюбленого);
* **система коментарів та вподобайок:** Можливість залишати, коментарі та ставити вподобайки контент;
* **персоналізовані рекомендації:** Отримання рекомендацій контенту на основі історії читання та вподобань;
* **керування профілем:** Редагування особистої інформації;
* **завантаження та публікація контенту:** Можливість завантажувати власні електронні книги, мангу, комікси у підтримуваних форматах.

**Функції які будуть доступні адміністраторам системи:**

**Управління контентом:** Модерація завантаженого контенту, схвалення/відхилення публікацій, редагування та видалення будь-якого контенту;

**Управління повідомленнями та скаргами:** Обробка скарг користувачів на контент або інших користувачів;

**Резервне копіювання та відновлення даних:** Забезпечення цілісності та доступності даних.

2.3 Характеристики користувачів

Таблиця 2.1 – Користувачі системи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Користувач** | **Головна цінність** | **Ставлення** | **Головний інтерес** | **Обмеження** |
| Розробник ПЗ | Автоматизація рутинних завдань, швидке створення каркасу проєкту | Позитивне. Очікується скорочення часу на розробку та зменшення помилок | Генерація якісного та підтримуваного коду | Потрібне правильне налаштування шаблонів та підтримка актуальних технологій |
| Технічний лідер/архітектор | Впровадження стандартизованого підходу до розробки проєктів | Позитивне. Очікується зниження технічного боргу та спрощення онбордингу нових розробників | Генерація коду відповідно до архітектурних принципів | Обмежена гнучкість у випадку нестандартних вимог |
| Проєктний менеджер | Скорочення часу на старт нових проєктів | Позитивне. Очікується підвищення продуктивності команди | Зниження витрат часу на початкову розробку | Необхідність навчання команди користуванню системою |

Кінець таблиці 2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Користувач** | **Головна цінність** | **Ставлення** | **Головний інтерес** | **Обмеження** |
| DevOps інженер | Стабільність та відповідність коду CI/CD процесам | Нейтральне. Очікується відповідність стандартам розгортання | Інтеграція з існуючими інструментами CI/CD | Потрібна сумісність із поточним стеком технологій |
| Клієнт (замовник ПЗ) | Зменшення витрат на розробку MVP | Позитивне. Очікується швидке отримання першої версії продукту | Прискорення процесу розробки | Обмежені можливості кастомізації без втручання розробника |

2.4 Загальні обмеження

При розробці програмної системи «Read and Feed» необхідно враховувати низку загальних обмежень, які можуть вплинути на проектування, реалізацію та розгортання системи:

**Технологічні обмеження:** Система буде розроблена з використанням сучасних веб-технологій та фреймворків (React.js для Front-end, Nest.js для Back-end), які забезпечують швидкість розробки, масштабованість та підтримуваність. Вибір конкретних версій та бібліотек буде обґрунтовано з урахуванням їх стабільності, популярності та наявності спільноти.

**Операційні обмеження:**

* **доступність та продуктивність:** Система повинна забезпечувати високу доступність (цілодобово, 7 днів на тиждень, за винятком планових технічних робіт) та прийнятну продуктивність, що дозволяє комфортно працювати значній кількості одночасних користувачів; Детальніші нефункціональні вимоги щодо продуктивності будуть визначені в розділі 3.2;
* **обсяг контенту:** Система повинна бути здатною обробляти та зберігати великі обсяги текстового та статичного візуального контенту (електронні книги, манга, комікси) з можливістю подальшого масштабування сховища;
* **модерація контенту:** Повинен бути реалізований механізм модерації завантаженого контенту для відповідності вимогам до вмісту та уникнення публікації неприпустимих матеріалів.

**Безпекові обмеження:**

* **захист даних:** Забезпечення конфіденційності, цілісності та доступності даних користувачів та контенту. Усі чутливі дані повинні передаватися та зберігатися в зашифрованому вигляді;
* **автентифікація та авторизація:** Надійна система автентифікації користувачів та авторизації доступу до ресурсів;
* **захист від поширених загроз:** Система повинна бути захищена від відомих веб-вразливостей, таких як SQL-ін'єкції та інших.

**Юридичні та етичні обмеження:**

* **авторське право:** Система повинна передбачати механізми для дотримання авторських прав на контент, включаючи можливість подачі скарг на порушення авторських прав.

2.5 Припущення й залежності

Цей розділ описує припущення, які були зроблені під час формування вимог, та залежності, що можуть вплинути на розробку та функціонування системи «Read and Feed».

**Припущення:**

* **наявність стабільного інтернет-з'єднання:** Припускається, що кінцеві користувачі матимуть стабільне та достатньо швидке інтернет-з'єднання для комфортного використання веб-платформи та завантаження контенту;
* **технічна грамотність користувачів:** Припускається, що користувачі мають базові навички роботи з веб-браузерами та цифровими платформами;
* **доступність контенту:** Припускається, що автори та правовласники будуть готові надавати контент для публікації на платформі відповідно до умов використання;
* **достатні обчислювальні ресурси:** Припускається, що для розгортання та підтримки системи будуть надані адекватні серверні ресурси (процесорний час, пам'ять, пропускна здатність мережі, місце на диску);
* **зацікавленість у багатомовному контенті:** Припускається, що існує достатній попит на контент українською та іншими мовами, що обґрунтовує інвестиції у багатомовну підтримку.

**Залежності:**

* **інструменти розробки та середовище:** Проект залежить від доступності та коректної роботи обраних інструментів розробки (IDE, компілятори, менеджери пакетів), а також сумісного середовища розробки;
* **база даних:** Функціонування системи залежить від наявності та стабільної роботи обраної СУБД (Postgress);
* **доступність контенту від правовласників:** Для наповнення платформи різноманітним та якісним контентом необхідне укладання угод або отримання дозволів від авторів та видавців.

**3 КОНКРЕТНІ ВИМОГИ**

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

Інтерфейс програмної системи реалізовано як односторінковий вебзастосунок з використанням React.js. Структура проєкту побудована на принципі розділення компонентів за сторінками: кожна сторінка має власні компоненти, що відповідають за конкретну функціональність. Це забезпечує зручну підтримку та розширюваність коду, а також дозволяє точно контролювати функціональне наповнення кожного розділу.

Для реалізації візуальних елементів використовувалась бібліотека Shadcn, яка надала готові компоненти, такі як кнопки, таблиці, діалогові вікна та форми. Усі повторно використовувані елементи були винесені до окремого модуля shared, що дозволило уникнути дублювання та забезпечити централізоване оновлення компонентів при зміні вимог до інтерфейсу.

Під час розробки основна увага приділялась реалізації повного функціоналу для користувачів, які можуть читати та завантажувати контент на платформу. Інтерфейс повинен підтримувати роботу всі трьох типів користувачів: неавторизований, авторизований та адміністратор. Інтерфейс створювався з увагою до адаптивності, але основними пристроями для тестування були комп’ютери з широкоформатними моніторами.

3.1.2 Апаратний інтерфейс

Мінімальні вимоги до апаратного інтерфейсу:

* доступ до мережі Інтернет (wifi або мобільний інтернет);
* операційна система Windows 10 та вище або MacOS для ноутбуків та IOS або Android для смарфонів;
* актуальний браузер (Google Chrome, Opera, Safari, Firefox, Microsoft Edge).

3.1.3 Програмний інтерфейс

Програмний інтерфейс системи реалізований у вигляді REST API, що забезпечує обмін даними між клієнтською та серверною частинами. Користувач може надсилати дані у форматі JSON. Серверна частина обробляє запити, динамічно створює необхідні сутності, оброблює їх, та відсилає відповідь в зручному та оптимізованому форматі. API захищене за допомогою JWT-автентифікації. Доступ до даних реалізовано через ORM Prisma, що забезпечує зручну взаємодію з реляційною СУБД.

3.1.4 Комунікаційний протокол

Комунікація між різними компонентами системи здійснюватиметься через протоколи HTTP/HTTPS. Архітектура побудована за клієнт-серверним принципом, де сервер виконує основні функції з управління ресурсами та надання послуг, які використовуються клієнтською частиною.

3.1.5 Обмеження пам’яті

Система не потребує значних обсягів оперативної пам’яті для своєї роботи, проте для стабільної генерації проєктів середньої складності рекомендується мінімум 4 ГБ оперативної пам’яті на сервері. У випадку обробки великих конфігурацій або паралельної роботи декількох користувачів можливе тимчасове збільшення споживання пам’яті до 8 ГБ і більше.

Клієнтська частина є вебзастосунком і не накладає суттєвих вимог до пам’яті на стороні користувача, окрім стандартних вимог сучасного браузера. Для забезпечення швидкодії і уникнення перевантаження пам’яті система використовує механізми оптимізації, такі як очищення тимчасових файлів після генерації, обмеження розміру конфігурацій, а також потокову обробку даних.

3.1.6 Операції

Взаємодія між клієнтом і серверною частиною системи здійснюється через HTTP-протокол. Клієнт надсилає запит до API, після чого отримує відповідь у форматі, що відповідає запиту (JSON). Для доступу до ресурсів використовуються URL-адреси, які слугують точками входу до відповідних функціональних можливостей системи.

Кожен URL-ресурс супроводжується HTTP-методом, який визначає тип операції, що виконується над ресурсом. Основні операції:

* GET – використовується для отримання інформації або даних з сервера;
* POST – дозволяє створювати нові ресурси;
* PUT – забезпечує повне оновлення наявного ресурсу;
* PATCH – виконує часткове оновлення даних;
* DELETE – видаляє вказаний ресурс із системи.

3.2 Властивості програмного продукту

Програмний продукт побудований із використанням сучасного технологічного стеку, що забезпечує швидкість розробки, масштабованість і зручність супроводу. Основний акцент зроблено на модульність і гнучкість: система підтримує генерацію програмних проєктів на основі шаблонів або розмітки, динамічне створення компонентів (моделі, сервіси, контролери), а також можливість розширення за рахунок підключення нових фреймворків. Архітектура побудована за принципами чистого коду та з використанням REST API, що гарантує простоту інтеграції з іншими системами. Продукт орієнтований як на індивідуальне використання, так і на командну розробку, з перспективою подальшого розширення функціональності.

3.3 Атрибути програмного продукту

3.3.1 Надійність

Програмна система повинна мати можливість відновлення з негативного стану та бути доступною для користувачів 99% часу.

3.3.2 Масштабованість

Програмна система повинна мати відкриті точки розширення для подальшого легкого розширення. Точки розширення мають дозволяти створення атомарних одиниць розширення для мінімальної зміни вже існуючої системи.

3.3.3 Відкрита до відлагодження

Система повинна мати можливість збору даних про свій стан для подальшого їх аналізу розробниками або аналітиками. При помилці, система повинна надавати вичерпний опис помилки з даними, яких повинно вистачити для ідентифікації такої самої помилки в майбутньому та її відтворення для подальшого вирішення.

3.3.4 Адаптивність

Інтерфейс системи повинен бути спроєктований для досягнення адаптивності на різних екранах, а саме на мобільних пристроях, планшетах та комп’ютерах.

3.3.5 Швидкодія

Система повинна підтримувати механізми збереження даних для подальшого їх перевикористання при потребі здобуття цих даних шляхом високонавантажених дій.

3.3.6 Підтримка багатьох браузерів

Програмна система має бути доступною та працювати однаковим чином на всіх сучасних браузерах.

3.3.7 Безпека даних

Програмна система повинна зберігати лише необхідну для ідентифікації користувача приватну інформацію, мінімізуючи обсяг персональних даних. Уся критична інформація зберігається в зашифрованому або захешованому вигляді. Для автентифікації та авторизації користувачів застосовується JWT (JSON Web Token), що дозволяє реалізувати безпечну безстейтову взаємодію між клієнтом і сервером, а також ефективно контролювати доступ до ресурсів системи.

3.3.8 Робота з одночасним доступом

Програмна система повинна мати можливість обробки паралельних запитів без впливу запитів на результат один одного, якщо це не пряма дія спрямована на зміну стану системи.

3.3.9 Захист від помилок

Частини інтерфейсу програмної системи повинні бути доступними лише при достатній кількості введених даних для запобігання подальших помилок при обробці цих даних.

3.4 Вимоги бази даних

У процесі розробки системи було обрано систему управління базами даних Postgress, оскільки вона забезпечує високу продуктивність, добру масштабованість, а також має відмінну сумісність із середовищем розробки. Крім того, ця СУБД гарантує надійне та безпечне зберігання даних.

Для реалізації моделі даних у межах проєкту було використано реляційний підхід, що передбачає представлення об'єктів предметної області у вигляді таблиць з можливістю встановлення між ними зв’язків через унікальні ідентифікатори — первинні та зовнішні ключі. Така структура дозволяє ефективно зберігати та обробляти структуровану інформацію, що позитивно впливає на загальну продуктивність системи. Для взаємодії з базою даних було обрано ORM-фреймворк Prisma. Його застосування спрощує доступ до даних за рахунок об’єктно-орієнтованого підходу, забезпечує захист від SQL-ін’єкцій та знижує ймовірність помилок завдяки використанню дерев виразів при формуванні запитів.