Специфікація програмного забезпечення

Вебсистема для контролю роботи зоомагазину

Software Requirements Specification

1.0

10.04.2025

Білозуб Данило Владиславович

**ІСТОРІЯ ЗМІН**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Опис** | **Автор** | **Коментарі** |
| 10.04.2025 | Створено пункти 1.1 - 1.5 | Білозуб Данило Владиславович |  |
| 12.04.2025 | Створено пункти 2.1 - 2.5 | Білозуб Данило Владиславович |  |
| 17.04.2025 | Створено пункти 3.1 - 3.4 | Білозуб Данило Владиславович |  |
| 27.04.2025 | Створено пункт 3.5 | Білозуб Данило Владиславович |  |

**ЗАТВЕРДЖЕННЯ ДОКУМЕНТУ**

Наступну специфікацію вимог до програмного забезпечення було прийнято та схвалено:

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

**ЗМІСТ**

[ВСТУП 4](#_Toc201159379)

[1.1 Огляд продукту 4](#_Toc201159380)

[1.2 Мета 4](#_Toc201159381)

[1.3 Межі 4](#_Toc201159382)

[1.4 Посилання 5](#_Toc201159383)

[1.5 Означення та абревіатури 6](#_Toc201159384)

[2.1 Перспективи продукту 7](#_Toc201159385)

[2.2 Функції продукту 8](#_Toc201159386)

[2.3 Характеристики користувачів 9](#_Toc201159387)

[2.4 Загальні обмеження 10](#_Toc201159388)

[2.5 Припущення й залежності 10](#_Toc201159389)

[3.1 Вимоги до зовнішніх інтерфейсів 11](#_Toc201159390)

[3.1.1 Інтерфейс користувача 11](#_Toc201159391)

[3.1.2 Апаратний інтерфейс 12](#_Toc201159392)

[3.1.3 Програмний інтерфейс 12](#_Toc201159393)

[3.1.4 Комунікаційний протокол 12](#_Toc201159394)

[3.1.5 Обмеження пам’яті 13](#_Toc201159395)

[3.1.6 Операції 13](#_Toc201159396)

[3.1.7 Функції продукту 13](#_Toc201159397)

[3.1.8 Припущення й залежності 14](#_Toc201159398)

[3.2 Властивості програмного продукту 14](#_Toc201159399)

[3.3 Атрибути програмного продукту 14](#_Toc201159400)

[3.3.1 Надійність 14](#_Toc201159401)

[3.3.2 Доступність 15](#_Toc201159402)

[3.3.3 Безпека 15](#_Toc201159403)

[3.3.4 Супроводжуваність 15](#_Toc201159404)

[3.3.5 Переносимість 15](#_Toc201159405)

[3.3.6 Продуктивність 16](#_Toc201159406)

[3.4 Вимоги бази даних 16](#_Toc201159407)

[3.5 Інші вимоги 16](#_Toc201159408)

1. ВСТУП

## 1.1 Огляд продукту

Запропонований програмний продукт є вебсистемою для управління зоомагазином, що поєднує функції обліку товарів, продажів, персоналу, залишків на складі та ведення інформації про тварин. Система також надає клієнтам можливість переглядати наявність тварин, їх опис і доступність у магазинах, а за потреби — забронювати тварину онлайн.

Вона орієнтована на малий і середній зообізнес, де часто відсутні спеціалізовані облікові рішення, що створює труднощі з інвентаризацією, контролем зарплат, каси та загальною звітністю. Продукт дозволяє централізувати управління всіма аспектами зоомагазину в одній системі, забезпечуючи гнучке налаштування ролей (адміністратор, працівник, клієнт) та зручний доступ до функцій через вебінтерфейс.

Цільова аудиторія: власники зоомагазинів, менеджери, продавці, працівники складу, а також кінцеві клієнти, які шукають тварин онлайн.

## 1.2 Мета

Мета роботи полягає у створенні вебплатформи, яка забезпечить повноцінну автоматизацію операцій зоомагазину. Система має надати можливості для:

* ведення обліку тварин та їх паспортних даних;
* інвентаризації товарів із фіксацією цін закупівлі та продажу;
* управління персоналом;
* оформлення продажів через зручний POS-інтерфейс;
* автоматичного підрахунку каси, зарплат та прибутків;
* формування щоденних та періодичних фінансових звітів;
* перегляду каталогу тварин з можливістю онлайн-бронювання клієнтом.

Впровадження такого рішення покликане не лише підвищити ефективність внутрішніх бізнес-процесів, а й покращити взаємодію з клієнтами та розширити цифрову присутність магазину.

## 1.3 Межі

Програмна система охоплює:

* бекенд-сервер, реалізований на Python 3.11 з використанням FastAPI;
* клієнтську частину на Vue.js 3 з підтримкою SPA;
* реляційну базу даних PostgreSQL для зберігання основних даних (тварини, товари, працівники, продажі тощо);
* захист API за допомогою JWT-аутентифікації та RBAC-доступу;
* Docker-контейнери для окремих компонентів системи (api, db, nginx, redis);
* базову звітність у вигляді щоденних PDF-звітів з касою та прибутком;
* можливість майбутньої інтеграції з мобільними клієнтами.

Система не охоплює функції онлайн-оплати, не виконує облік фіскальної документації та не має вбудованого календаря для медичних записів тварин.

## 1.4 Посилання

* ISO/IEC 25010 – Якісні характеристики ПЗ: https://www.iso.org/standard/35733.html
* RFC 7519 – JSON Web Token (JWT): <https://datatracker.ietf.org/doc/html/rfc7519>
* FastAPI Docs – https://fastapi.tiangolo.com/
* Vue.js 3 Docs – <https://vuejs.org/>
* PostgreSQL Docs – <https://www.postgresql.org/docs/>
* SQLAlchemy 2 Docs – https://docs.sqlalchemy.org/en/20/
* Docker Compose Reference – https://docs.docker.com/compose/
* GitHub Actions – <https://docs.github.com/en/actions>
* Alembic Docs – https://alembic.sqlalchemy.org/en/latest/

## 1.5 Означення та абревіатури

Означення:

* Тварина – об’єкт у системі, що включає породу, стать, вік, опис, фото, медичні дані.
* Товар – будь-яка продукція (корм, аксесуари), яку можна додати в магазин.
* Працівник – користувач, який виконує операції продажу або обліку.
* Продаж – транзакція, яка включає одну або кілька позицій товару/тварини.
* POS – інтерфейс для швидкого оформлення продажу працівником.
* Звіт – автоматично сформований файл, який підсумовує діяльність магазину за день.

Абревіатури:

* UI (User Interface) – користувацький інтерфейс
* API (Application Programming Interface) – інтерфейс прикладного програмування
* DB (Database) – база даних
* JWT (JSON Web Token) – токен аутентифікації
* RBAC (Role-Based Access Control) – контроль доступу на основі ролей
* SPA (Single Page Application) – односторінковий вебзастосунок
* ORM (Object Relational Mapping) – об’єктно-реляційне відображення
* CRUD (Create, Read, Update, Delete) – базові операції з даними
* PDF (Portable Document Format) – формат електронного документа
* CI/CD (Continuous Integration / Continuous Deployment) – безперервна інтеграція та доставка

2. ЗАГАЛЬНИЙ ОПИС

## 2.1 Перспективи продукту

Розробка цифрової системи для зоомагазину відкриває значні можливості для автоматизації щоденних бізнес-процесів малого та середнього роздрібного сегмента. У таких магазинах часто відсутні централізовані облікові рішення — операції ведуться в Excel, на папері або в месенджерах, що унеможливлює єдиний контроль над персоналом, тваринами, товарами та продажами. Запропонована вебсистема дозволяє об’єднати весь облік у єдиній інтерфейсній платформі, доступній онлайн з будь-якого пристрою.

Система охоплює модулі обліку тварин з паспортами, медичною історією, фото та статусами вакцинації, а також інтегрований облік товарів (корм, ліки, аксесуари) та складських залишків. Важливою перевагою є реалізація POS-функціоналу з інтуїтивним інтерфейсом для продавця — що забезпечує швидке оформлення продажів, підрахунок виручки та автоматичне оновлення складу.

Також система відкриває новий рівень взаємодії з клієнтами. Завдяки онлайн-кабінету покупець може переглядати наявних тварин, ознайомлюватися з їх характеристиками та забронювати улюбленця на визначений термін.

Цифрова платформа забезпечує контроль над персоналом — фіксує продажі за працівниками, обчислює зарплати згідно з тарифами та дозволяє керівництву формувати звіти за день, тиждень або місяць. У перспективі система може бути інтегрована з CRM, онлайн-оплатами або навіть мікрочипами тварин для відстеження медичного статусу.

## 2.2 Функції продукту

Система підтримує широкий набір функцій, що охоплюють як внутрішню адміністративну частину, так і клієнтський інтерфейс:

* Облік тварин із паспортними даними, фото, медичними мітками та станом (активна, заброньована, продана).
* Управління товарами: додавання, редагування, категоризація, фото, оптова/роздрібна ціна.
* Продажі: касовий інтерфейс з формуванням чеків, підрахунком каси, онлайн-звітуванням.
* Профілі працівників із можливістю призначення ролей та фіксації продажів.
* Ролі користувачів: адміністратор, касир, складський менеджер, гість, покупець.
* Панель адміністратора з графіками продажів, популярності товарів, обсягів виручки.
* Клієнтський сайт: каталог тварин, перегляд фото/інформації, функція онлайн-бронювання.

Усі функції доступні з веббраузера, що забезпечує доступність з ПК, планшетів чи смартфонів.

## 2.3 Характеристики користувачів

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

* Гість — користувач без авторизації. Може переглядати тварин, здійснювати пошук та оформити реєстрацію.
* Покупець — авторизований користувач. Має змогу переглядати тварин, подавати запити на бронювання, переглядати історію своїх дій.
* Працівник магазину — має доступ до POS-системи, може створювати продажі, переглядати залишки, працювати з клієнтами.
* Менеджер магазину — керує товарами, тваринами, персоналом і контролює продажі.
* Адміністратор — глобальний контроль над усіма магазинами, формує звіти, змінює ролі користувачів, має доступ до внутрішніх аналітичних функцій.

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

## 2.4 Загальні обмеження

При розробці та впровадженні системи слід враховувати низку обмежень:

* Продуктивність клієнтських пристроїв. Повноцінна робота платформи потребує сучасного браузера з підтримкою JavaScript. На слабких пристроях можливе зниження швидкості відтворення таблиць або рендерингу.
* Мережева залежність. Усі операції виконуються через API, тому стабільне інтернет-з’єднання є критичним.
* Обмеження у файловому сховищі. Завантаження великих фото або обробка масивів зображень можуть потребувати розширення обсягу MinIO або обмеження на розмір файлів.
* Відсутність мобільного застосунку. Наразі реалізовано лише адаптивний вебінтерфейс.
* Локалізація. Система доступна українською мовою; підтримка багатомовності планується у наступних релізах.

## 2.5 Припущення й залежності

* Браузерна підтримка. Платформа повинна підтримувати останні версії Chrome, Firefox, Edge. Тестування на Safari або старих Android WebView не гарантує повної сумісності.
* Інтеграція з електронною поштою. Для надсилання запрошень/бронювань використовується SMTP-сервер; відмова сервера блокує повідомлення.
* Залежність від PostgreSQL. Уся бізнес-логіка побудована на реляційних запитах. Перехід на інші СУБД потребуватиме значної адаптації ORM.
* Алгоритми обчислення зарплати та каси. Розрахунки залежать від вхідних параметрів — тарифів, ролей, кількості змін.
* Контейнерна інфраструктура. Розгортання базується на Docker Compose. У випадку хостингу без підтримки контейнерів потрібне ручне налаштування залежностей.
* Обмеження API-запитів. Частина запитів (пошук, фільтрація) використовує raw-SQL. Масштабування потребуватиме оптимізації або застосування кешування (наприклад, Redis).
* Вимоги до безпеки. JWT-аутентифікація використовується для захисту доступу. Зміни в стандартах шифрування можуть потребувати оновлення middleware.

3. КОНКРЕТНІ ВИМОГИ

## 3.1 Вимоги до зовнішніх інтерфейсів

### 3.1.1 Інтерфейс користувача

Інтерфейс побудовано на основі Vue.js 3 з використанням сучасних принципів SPA. Користувачі взаємодіють із системою через адаптивний вебінтерфейс, який оптимізований як для десктопних, так і мобільних пристроїв. Інтерфейс підтримує рольову модель: адміністратор, працівник, власник, покупець. Основні компоненти включають дашборд магазину, форму продажів, управління працівниками, перегляд тварин і бронювання. Особлива увага приділена зручності роботи касира — реалізовано POS-механізм з формуванням кошика та підтвердженням продажу в один клік.

### 3.1.2 Апаратний інтерфейс

Система не вимагає специфічного обладнання. Достатньо стандартного ПК або планшета з доступом до браузера (Chrome, Firefox, Safari). Для стабільної роботи клієнта потрібно не менше 2 ГБ ОЗП, а для серверної частини — не менше 4 ГБ ОЗП та SSD-накопичувач для швидкого доступу до бази даних і медіаконтенту (фото тварин тощо).

### 3.1.3 Програмний інтерфейс

Взаємодія між клієнтом і сервером реалізована через RESTful API на базі FastAPI. Усі сервіси типізовані через Pydantic. Для доступу до БД застосовується SQLAlchemy 2.0. Модулі CRUD логічно відокремлені. Інтеграція з зовнішніми сервісами (наприклад, email, Telegram, платіжні шлюзи) планується через окремі адаптери.

### 3.1.4 Комунікаційний протокол

Передача даних здійснюється виключно через HTTPS із підтримкою TLS 1.3. Авторизація відбувається за допомогою JWT-токенів. Усі зовнішні запити логуються. Передбачена підтримка CORS для безпечної взаємодії між фронтендом та API.

### 3.1.5 Обмеження пам’яті

Середнє споживання памʼяті на клієнтській стороні не перевищує 100 МБ. З боку серверної частини обмеження залежать від навантаження: PostgreSQL зберігає дані у вигляді індексованих таблиць.

### 3.1.6 Операції

Користувачі мають змогу здійснювати такі операції: реєстрація магазину, додавання товарів і тварин, створення продажу, керування залишками, формування звітів по обороту та зарплаті, запрошення працівників, бронювання тварин. Кожна операція перевіряється через систему прав доступу.

### 3.1.7 Функції продукту

Система забезпечує:

* Авторизацію та багаторольову ідентифікацію.
* Управління магазином (товари, тварини, працівники).
* Створення та підтвердження продажу через POS-інтерфейс.
* Ведення історії продажів і бронювань.
* Автоматичне нарахування зарплати.
* Генерацію щоденних/місячних звітів.
* Аналітику популярності товарів та продажів по годинах.

### 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.

## 3.2 Властивості програмного продукту

Програмне забезпечення має розгалужену логіку керування зоомагазином із урахуванням нюансів продажу тварин, товарів, та обліку праці. Воно здатне функціонувати як у локальному магазині, так і в мережевій структурі з кількома магазинами та віддаленим адмініструванням. Завдяки модульному підходу підтримується гнучке масштабування і додавання нових функцій (наприклад, веткарти, клієнтські кабінети, мобільний додаток).

## 3.3 Атрибути програмного продукту

### 3.3.1 Надійність

Система реалізована з урахуванням обробки винятків, резервного копіювання БД через cron. Розгортання розраховано на безперервну роботу 24/7 з автоматичним перезапуском контейнерів при збої.

### 3.3.2 Доступність

Доступ до системи можливий з будь-якого браузера. Завдяки контейнеризації забезпечується швидке масштабування та розподіл навантаження.

### 3.3.3 Безпека

Передбачено валідацію запитів, обмеження прав доступу, JWT-аутентифікацію, CORS. Дані шифруються при передачі. Фото зберігаються в окремому об’єктному сховищі MinIO.

### 3.3.4 Супроводжуваність

Код написано у відповідності до стандартів PEP8. CI/CD налаштовано через GitHub Actions. Підтримується автоматичне розгортання і тестування при пуші.

### 3.3.5 Переносимість

Увесь стек працює в Docker, що забезпечує незалежність від ОС. Платформа може бути перенесена на інший хостинг із мінімальними змінами.

### 3.3.6 Продуктивність

Система оптимізована для обробки до 100 одночасних користувачів. Основні запити кешуються. Для інтенсивних запитів (залишки, звіти) використано матеріалізовані представлення.

## 3.4 Вимоги бази даних

База даних PostgreSQL зберігає інформацію про користувачів, магазини, продажі, залишки, тварин, зарплати. Передбачено логування подій. Усі ключові поля проіндексовані. Для найчастіших запитів реалізовано materialized views. Міграції здійснюються через Alembic.

## 3.5 Інші вимоги

Захист даних. Усі персональні дані користувачів захищені згідно з GDPR. Користувач має право на видалення облікового запису.

Правові обмеження. Система не дозволяє публікацію непридатного або незаконного контенту. Додавання тварини можливе лише з обов’язковими параметрами — дата народження, наявність веткарти тощо.

Енергоефективність. Використання контейнерів та інструментів моніторингу дозволяє контролювати навантаження на інфраструктуру. Пріоритет — зниження енергоспоживання через оптимізацію запитів.