Програмна система для обліку товарів на складі. IoT система для зручного підрахунку товарів

Специфікація вимог до програмного забезпечення

1.0

22.06.2025

Черняк Андрій

**Історія версій**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Опис** | **Автор** | **Коментар** |
| 22.06.25 | Версія 1.0 | Черняк Андрій | Створення документу |

**Зміст**

[1. Вступ 4](#_Toc201532961)

[1.1 Огляд продукту 4](#_Toc201532962)

[1.2 Мета 4](#_Toc201532963)

[1.3 Межі 5](#_Toc201532964)

[1.4 Посилання 5](#_Toc201532965)

[1.5 Означення та абревіатури 6](#_Toc201532966)

[2. Загальний опис 6](#_Toc201532967)

[2.1 Перспективи продукту 6](#_Toc201532968)

[2.2 Функції продукту 7](#_Toc201532969)

[2.3 Характеристики користувачів 8](#_Toc201532970)

[2.4 Загальні обмеження 8](#_Toc201532971)

[2.5 Припущення та залежності 9](#_Toc201532972)

[3. Конкретні вимоги 10](#_Toc201532973)

[3.1 Вимоги до зовнішніх інтерфейсів 10](#_Toc201532974)

[3.1.1 Інтерфейс користувача 10](#_Toc201532975)

[3.1.2 Апаратний інтерфейс 14](#_Toc201532976)

[3.1.3 Програмний інтерфейс 16](#_Toc201532977)

[3.1.4 Комунікаційний протокол 16](#_Toc201532978)

[3.1.5 Обмеження пам’яті 16](#_Toc201532979)

[3.1.6 Операції 16](#_Toc201532980)

[3.1.7 Функції продукту 16](#_Toc201532981)

[3.2 Властивості програмного продукту 17](#_Toc201532982)

[3.3 Атрибути програмного продукту 17](#_Toc201532983)

[3.3.1 Надійність 17](#_Toc201532984)

[3.3.2 Доступність 18](#_Toc201532985)

[3.3.3 Безпека 18](#_Toc201532986)

[3.3.4 Супровід 18](#_Toc201532987)

[3.3.5 Переносимість 18](#_Toc201532988)

[3.3.6 Продуктивність 18](#_Toc201532989)

[3.4 Вимоги до бази даних 18](#_Toc201532990)

[3.5 Інші вимоги 19](#_Toc201532991)

[4. Додаткові матеріали 20](#_Toc201532992)

[4.1 Діаграми баз даних 20](#_Toc201532993)

[4.2 Діаграма прецедентів 21](#_Toc201532994)

# 1. Вступ

## 1.1 Огляд продукту

CRM-система для мережі магазинів — це веб-програмна платформа, яка об'єднує веб-застосунок (React.js), серверну частину (Express.js) та IoT-компонент (емулятор зчитувача штрих-кодів на базі Wokwi). Система дозволяє управляти магазинами, акціями, клієнтськими зверненнями (тікетами) та тренінгами для персоналу з єдиного інтерфейсу. IoT-компонент забезпечує швидке зчитування товарів та перевірку наявності на складі в режимі реального часу. Продукт спрямований на автоматизацію рутинних процесів, покращення обслуговування клієнтів та підвищення ефективності бізнес-операцій в умовах високої конкуренції на роздрібному ринку.

## 1.2 Мета

Метою розробки програмної CRM-системи є створення інтегрованої, масштабованої та зручної платформи для ефективного управління мережею роздрібних магазинів. Система повинна забезпечити централізоване керування клієнтськими зверненнями, акціями, навчанням персоналу та залишками на складах. Інтеграція з IoT-компонентом дозволить отримувати інформацію про наявність товарів у режимі реального часу. Проєкт також спрямований на покращення взаємодії з клієнтами, підвищення рівня обслуговування, зменшення кількості помилок та автоматизацію рутинних бізнес-процесів. Система має забезпечити зручний інтерфейс для працівників, швидке реагування на звернення клієнтів та підтримку прийняття управлінських рішень на основі аналітичних даних.

## 1.3 Межі

Програмна CRM-система для мережі магазинів включає веб-додаток, який забезпечує основний функціонал: управління клієнтськими зверненнями (тікет-система), створення та контроль акцій, перегляд і навчання персоналу, а також відображення наявності товарів на складах завдяки інтеграції з IoT-компонентом (зчитувач штрих-кодів). Система орієнтована на внутрішнє використання працівниками компанії, тому не передбачає окремого мобільного застосунку для клієнтів.

Серверна частина реалізована за допомогою Node.js (REST API) з розділенням модулів за функціональністю. Клієнтська частина — на React з використанням компонентного підходу.

## 1.4 Посилання

Текст документу містить наступні посилання:

1. Посилання на репозиторії проєкту:

https://github.com/NureCherniakAndrii/pzpi-21-9-cherniak-andrii-bachelor-work

## 1.5 Означення та абревіатури

В документі використовуються наступні означення та абревіатури:

HTTP – HyperText Transfer Protocol

REST – REpresentational State Transfer

SQL – Structured Query Language

# 2. Загальний опис

## 2.1 Перспективи продукту

У зв’язку з постійним зростанням конкуренції в роздрібній торгівлі, підвищенням очікувань клієнтів щодо сервісу та необхідністю цифровізації бізнес-процесів, впровадження CRM-системи для мережі магазинів є актуальним і стратегічно важливим кроком. Автоматизація обслуговування, обліку звернень, керування акціями та навчанням персоналу дозволяє суттєво підвищити ефективність роботи компанії та якість взаємодії з клієнтами.

Розроблена CRM-система орієнтована на такі напрямки розвитку:

1. **Мережі роздрібної торгівлі** — централізоване керування всіма точками продажу, єдина база клієнтів та прозора аналітика.
2. **Покращення клієнтського сервісу** — швидкий облік звернень, історії покупок і персоналізовані пропозиції.
3. **Оптимізація роботи персоналу** — управління навчанням, контроль виконання завдань та оцінка ефективності.
4. **Управління акціями** — створення та контроль знижок, акцій та бонусних програм в реальному часі.
5. **Інтеграція з IoT** — автоматичне оновлення інформації про наявність товарів на складах і зменшення ризику людських помилок.
6. **Масштабування** — можливість адаптації системи для нових магазинів, регіонів чи навіть виходу на інші ринки.

Таким чином, система забезпечує довгострокові переваги для бізнесу, сприяючи цифровій трансформації та підвищенню конкурентоспроможності

## 2.2 Функції продукту

CRM-система для мережі магазинів є централізованою веб-платформою для управління клієнтськими зверненнями, персоналом, акціями та процесами навчання. Основний функціонал включає:

* **Управління клієнтами** — створення, перегляд і редагування карток клієнтів, історія звернень та покупок.
* **Облік звернень** — реєстрація вхідних запитів, їх статус, відповідальні співробітники, історія взаємодій.
* **Навчання персоналу** — створення курсів/інструкцій, тестування, перегляд результатів, зручний інтерфейс для учнів.
* **Керування акціями** — створення маркетингових кампаній, додавання опису, дати, умов участі, відображення на фронтенді.
* **Панель адміністратора** — налаштування доступів, ролей (менеджер, адміністратор, оператор), перегляд статистики.
* **Система повідомлень** — внутрішні повідомлення про нові звернення, завершення навчання, старт акцій.
* **Аналітика та звітність** — графіки продажів, ефективність персоналу, звіти за періодами.
* **Безпека та аутентифікація** — доступ лише через захищену авторизацію з ролями, ведення логів дій.
* **Інтерфейс керування мережею** — підтримка кількох магазинів із незалежною статистикою та доступами.
* **Сумісність з іншими системами** — можливість інтеграції з POS, ERP або складським обліком через API.

Такий набір функцій дозволяє ефективно організувати роботу роздрібної мережі, забезпечуючи прозорість процесів, швидке обслуговування клієнтів і контроль за всіма ключовими аспектами бізнесу.

## 2.3 Характеристики користувачів

CRM-система взаємодіє з чотирма основними типами користувачів:

* **Адміністратор мережі** — має повний доступ до всіх магазинів, аналітики, користувачів, звернень, навчальних модулів та акцій. Несе відповідальність за управління ролями, перегляд глобальної статистики та конфігурацію системи.
* **Менеджер магазину** — керує локальним персоналом, переглядає звернення клієнтів, проводить навчання співробітників, активує локальні акції.
* **Оператор** — відповідає за прийом звернень, комунікацію з клієнтами, оновлення даних у CRM, передачу завдань менеджеру.

Кожна роль має власний рівень доступу до функціоналу, сформований згідно принципів **least privilege**.

## 2.4 Загальні обмеження

CRM-система для мережі магазинів реалізована як веб-застосунок із трирівневою архітектурною моделлю, що включає:

* **Рівень презентації (Presentation Layer)** — інтерфейс користувача, реалізований за допомогою React.js, забезпечує зручну та адаптивну взаємодію з системою для менеджерів, операторів і адміністраторів.
* **Рівень бізнес-логіки (Application/Logic Layer)** — REST API, створене на базі Node.js з використанням Express.js, обробляє запити клієнтів, реалізує правила бізнес-процесів та координує роботу з даними.
* **Рівень даних (Data Layer)** — PostgreSQL використовується як основна база даних, оптимізована для великого навантаження та транзакційної обробки.

Основні технічні обмеження:

* **База даних** — PostgreSQL, оптимізована для роботи з великими обсягами записів і транзакцій.
* **Безпека** — автентифікація реалізована через JWT з контролем доступу на основі ролей.
* **Інтерфейс користувача** — адаптивна верстка на базі Tailwind CSS, з урахуванням специфіки роботи різних ролей користувачів.
* **Поточні обмеження**:
  + Відсутність мобільного додатку.
  + Доступ до адміністративного функціоналу можливий лише через веб-версію.
  + Відсутня підтримка офлайн-режиму.

Перспективи розвитку:

У подальшому планується перехід до мікросервісної архітектури з оркестрацією за допомогою Docker і Kubernetes для покращення масштабованості, відмовостійкості та гнучкості оновлень окремих компонентів.

## 2.5 Припущення та залежності

Припущення

1. Існує стійкий попит на інструменти для менторства, індивідуального навчання та управління взаємодією з клієнтами в цифровому середовищі.
2. Цільова аудиторія володіє базовими навичками цифрової грамотності, достатніми для користування веб-платформою.
3. Інтегровані сторонні сервіси (зокрема Stripe та Talk.js) будуть стабільно працювати протягом життєвого циклу платформи.
4. Команда проєкту має достатній рівень технічної компетентності для впровадження та підтримки всіх запланованих функцій.

Залежності

1. Серверна частина:
   * REST API — реалізація за допомогою Node.js.
   * PostgreSQL — база даних для зберігання структурованої інформації.
   * JWT – автентифікація для застосунку.
2. Клієнтська частина:
   * React.js — бібліотека для веб-додатку.
   * Tailwind CSS — адаптивна верстка.
3. CI/CD та інфраструктура:
   * GitHub — зберігання проекту.
4. Технічні умови експлуатації:
   * Продуктивність платформи залежить від стабільності інтернет-з'єднання як користувача, так і хостинг-середовища.
   * Надійність функціоналу напряму пов'язана зі стабільною роботою сторонніх API та сервісів.
5. Організаційні фактори:
   * Успішне впровадження нових модулів залежить від наявного часу та ресурсів команди на вивчення технологій, розробку, тестування та інтеграцію.

# 3. Конкретні вимоги

## 3.1 Вимоги до зовнішніх інтерфейсів

### 3.1.1 Інтерфейс користувача

Спочатку працівнику треба увійти до системи, тому перша сторінка застосунку – це вхід (рис. 3.1). Для входу користувачу треба:

1. Написати свою пошту
2. Написати пароль
3. Натиснути на кнопку «Login»

Якщо у користувача немає облікового запису, треба повідомити адміністратора, який створить акаунт та пароль для користувача.

Изображение выглядит как снимок экрана, дизайн

Автоматически созданное описание

Рисунок 3.1 – Сторінка входу до системи

Після входу до системи користувача зустрічає головна сторінка (рис. 3.2).

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3.2 – Головна сторінка застосунку

Це також сторінка навігації, де користувач може перейти до наступних розділів:

* Навчання (Trening)
* Акції (Promotion)
* Магазин (Store)
* Тікет (Ticket)

При переході на сторінку тренінгів працівник побачить список тренінгів, де є можливість переглянути інформацію про тренінг, створити новий тренінг, редагувати існуючі тренінги та видалити існуючі тренінги (рис. 3.3).

Изображение выглядит как снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 3.3 – Сторінка тренінгів (Training)

На сторінці тренінгів при натисканні на кнопку «Add» з’явиться модульна форма додавання тренінгів (рис. 3.4).

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

Рисунок 3.4 – Модульна форма додавання тренінгів

Для успішного додавання треба вписати такі дані:

* Назва (Title)
* Опис (Description)
* Тривалість (Duration)
* Вибрати працівників (Name)

Натиснувши на кнопку «Redact» відкриється модульна форма редагування тренінгу (рис. 3.5).

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 3.5 – Модульна форма редагування тренінгу

При виборі сторінки акцій перед користувачем з’являється сторінка акцій, де він може переглянути інформацію про акцію, створити нову акцію, редагувати існуючі акції та видалити існуючі акції. Також працівник може додати нові акції, натиснувши на кнопку «Add».

Статуси створюються вищими адміністраторами, бо це стосується усіх магазинів та усієї системи, тому вони приготовлені заздалегідь. Це схожа функціональність з популярною CRM-системою ServiceNow.

Також працівнику доступна можливість редагувати існуючі тренінги натиснувши на кнопку «Redact».

Натиснувши на кнопку сторінки магазинів перед користувачем з’явиться сторінка магазинів, де він має можливість переглянути інформацію про магазини, створити новий магазин, редагувати існуючі магазини та видалити існуючі магазини (рис. 3.6).

Изображение выглядит как текст, снимок экрана, еда

Автоматически созданное описание

Рисунок 3.6 – Сторінка магазинів

Також працівник має можливість додавати нові магазини, відредагувати існуючі магазини, натиснувши на кнопку «Redact».

Перейшовши до сторінки тікетів користувач бачить список усіх тікетів та має можливість переглянути інформацію про тікети, створити новий тікет, редагувати існуючі тікети та видалити існуючі тікети.

### 3.1.2 Апаратний інтерфейс

Працівник сканує штрих-код і на екран виводиться така інформація:

1. Назва продукту
2. Кількість цього продукту на складі магазину

Изображение выглядит как снимок экрана, текст, дизайн

Автоматически созданное описание

Рисунок 3.7 – Схема IoT-компоненту

Приклад роботи IoT-компоненту представлено на рис. 4.2

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 3.8 – Робота IoT-компоненту

Таким чином використання IoT у програмній системі для обліку товарів на складі з інтуїтивним інтерфейсом і ролевим поділом функціональності є ключовим елементом, що забезпечує її унікальність, ефективність і відповідність сучасним потребам роздрібної торгівлі.

### 3.1.3 Програмний інтерфейс

Система сумісна з усіма сучасними браузерами (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge). Платформа розроблена з використанням HTML5, CSS3, JavaScript ES6, React.js.

### 3.1.4 Комунікаційний протокол

Взаємодія між клієнтською та серверною частинами реалізується через HTTPS та REST API.

### 3.1.5 Обмеження пам’яті

Жодних специфічних обмежень пам’яті не встановлюється. Платформа працює у веб-браузері, і пам’ять визначається ресурсами пристрою користувача та налаштуваннями браузера.

### 3.1.6 Операції

Платформа реалізує REST-архітектуру з такими HTTP-методами:

* GET — отримання даних (профіль, оголошення, повідомлення)
* POST — створення (реєстрація, нове бронювання)
* PUT — редагування (профілю, послуги)
* DELETE — видалення (бронювання, оголошення)

### 3.1.7 Функції продукту

ГФ-1: Автентифікація користувачів  
ГФ-2: Створення, перегляд, редагування та видалення продуктів  
ГФ-3: Створення, перегляд, редагування та видалення акцій  
ГФ-4: Створення, перегляд, редагування та видалення магазинів  
ГФ-5: Створення, перегляд, редагування та видалення тікетів  
ГФ-6: Отримання кількості товарів по штрихкоду  
ГФ-7: Управління та перегляд користувачами

## 3.2 Властивості програмного продукту

* **Цільова аудиторія**  
  CRM-система орієнтована на менеджерів магазинів, операторів та адміністраторів розгалуженої торгової мережі.
* **Підключення до мережі**  
  Для роботи платформи обов’язкове стабільне інтернет-з’єднання; офлайн-режим не підтримується.
* **Продуктивність**  
  Система повинна забезпечувати час відповіді не більше **30 секунд** при роботі з великими обсягами даних (наприклад, облік товарів, звернення).
* **Безпека**  
  Впроваджено автентифікацію через JWT, а також контроль доступу на основі ролей для забезпечення безпеки даних.
* **Інтерфейс**  
  Веб-інтерфейс адаптивний, реалізований із застосуванням Tailwind CSS, забезпечує зручну роботу як на десктопах, так і на мобільних пристроях.

## 3.3 Атрибути програмного продукту

### 3.3.1 Надійність

* Система мусить повертати правильні дані тільки.
* При помилці видавати зрозуміле повідомлення.

### 3.3.2 Доступність

* Доступ до функціоналу мають лише авторизовані користувачі.
* Адміністративні функції доступні лише адміністраторам.

### 3.3.3 Безпека

* Обліковий запис захищений паролем.
* Доступ лише після вірного введення логіна і пароля.
* Функціонал доступний згідно з ролями.

### 3.3.4 Супровід

* Система має бути легко розширювана новими функціями.
* Компоненти системи (сервер, клієнт) відокремлені.

### 3.3.5 Переносимість

* Веб-додаток підтримується у сучасних браузерах.

### 3.3.6 Продуктивність

* Перехід між вкладками — до 1 секунди.

## 3.4 Вимоги до бази даних

У CRM системі PostgreSQL використовується як СУБД. Переваги:

* можливість ефективного масштабування системи;
* забезпечення надійності транзакцій відповідно до принципів ACID;
* розвинуті механізми управління правами доступу;
* підтримка складних взаємозв’язків між даними та ефективна індексація;
* безшовна інтеграція з FastAPI і SQLAlchemy.

База даних містить інформацію про користувачів, оголошення, категорії, сесії, транзакції, повідомлення, відгуки та адміністративні дані.

### 3.5 Інші вимоги

* До систему мають доступ тільки зареєстровані користувачі.
* Інтерфейс мусить бути швидким.
* Сервер мусить відповідати швидко.
* IoT система повинна повертати вірні дані.

# 4. Додаткові матеріали

## 4.1 Діаграми баз даних

Изображение выглядит как текст, диаграмма, План, Параллельный

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9 – ER-діаграма бази даних

## Изображение выглядит как текст, диаграмма, снимок экрана, Параллельный Автоматически созданное описание4.2 Діаграма прецедентів

Рисунок 10 – Use-Case діаграма