Software Requirements Specification

for

Сервіс для вирішення питань пов'язаних із замовленнями в піцерії

Version 1.0 approved

Prepared by Мартинов Богдан Валерійович

Харківський національний університет радіоелектроніки

16.04.2025

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 References 2

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 3

2.4 Role Management 3

2.5 Design and Implementation Constraints 3

2.6 User Documentation 3

2.7 Progress Monitoring 3

3. System Features 4

3.1 User Registration and Authentication 4

3.2 Student Profile Management 4

3.3 Academic Achievements Monitoring 5

3.4 Notification System 5

3.5 News Board Management 6

3.6 Data Import Functionality 6

3.7 Rating Table Generation 6

4. External Interface Requirements 7

4.1 User Interfaces 7

4.2 Hardware Interfaces 7

4.3 Software Interfaces 7

4.4 Communications Interfaces 7

5. Other Nonfunctional Requirements 8

5.1 Performance Requirements 8

5.2 Safety Requirements 8

5.3 Security Requirements 8

5.4 Software Quality Attributes 8

6. Other Requirements 8

Appendix A: Glossary 8

Appendix B: Analysis Models 9

Appendix C: Issues List 10

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Мартинов Богдан | 16.06.2025 | Initial creation of SRS document | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

Цей документ специфікації вимог до програмного забезпечення (SRS) описує вимоги до веб-застосунку «Сервіс для вирішення питань, пов’язаних із замовленнями в піцерії». Документ охоплює повний функціонал системи — від реєстрації користувачів і створення замовлень до інтеграції з платіжною системою LiqPay, відстеження статусу замовлень і системи сповіщень. Метою є чітке визначення функціональних і нефункціональних вимог для розробників, тестувальників та зацікавлених сторін.

## Document Conventions

У документі використовуються стандартні конвенції для чіткості та зрозумілості:

* вимоги позначені унікальними ідентифікаторами (наприклад, REQ-1, REQ-2);
* пріоритети вимог позначені як високий, середній або низький;
* жирний шрифт використовується для назв розділів і ключових термінів.

## Intended Audience and Reading Suggestions

Цей документ призначений для:

* розробників: для розуміння технічних вимог та реалізації системи;
* тестувальників: для створення тест-кейсів та перевірки відповідності вимогам;
* керівників проєкту: для оцінки обсягу робіт та планування;
* викладачів та адміністраторів: для розуміння функціоналу системи;
* студентів: для ознайомлення з інтерфейсом та можливостями.

Рекомендується розпочати з розділу 1 (Вступ) для загального розуміння, перейти до розділу 2 (Загальний опис) для контексту, а потім детально вивчити розділ 3 (Системні функції) для ознайомлення з основними можливостями. Розділи 4 та 5 важливі для технічних деталей реалізації та нефункціональних вимог.

## Project Scope

Веб-застосунок «Сервіс для вирішення питань, пов’язаних із замовленнями в піцерії» призначений для автоматизації всього циклу онлайн-замовлення піци. Система забезпечує:

* авторизацію та реєстрацію користувачів (email, Google OAuth);
* перегляд каталогу піц із фільтрацією, сортуванням і збереженням параметрів у URI;
* кастомізацію замовлень: вибір розміру та інгредієнтів;
* кошик: додавання, редагування та видалення позицій;
* оформлення замовлення з введенням адреси доставки;
* оплата онлайн через LiqPay;
* сповіщення електронною поштою про кожну зміну статусу;
* інтеграцію з Instagram Stories для просування акцій;
* адаптивний дизайн для десктопів і планшетів.

## References

* Next.js Documentation. URL: https://nextjs.org/docs (дата звернення: 06.05.2024)
* Prisma ORM Documentation. URL: https://www.prisma.io/docs (дата звернення: 06.05.2024)
* Vercel PostgreSQL. URL: https://vercel.com/docs/storage/vercel-postgres (дата звернення: 06.05.2024)
* Tailwind CSS Documentation. URL: https://tailwindcss.com/docs/installation (дата звернення: 06.05.2024)
* NextAuth.js Documentation. URL: https://next-auth.js.org/getting-started/introduction (дата звернення: 06.05.2024)
* Zustand State Management Documentation. URL: https://github.com/pmndrs/zustand (дата звернення: 06.05.2024)
* LiqPay Documentation. URL: https://www.liqpay.ua/doc (дата звернення: 06.05.2024)
* Instagram Graph API Documentation. URL: https://developers.facebook.com/docs/instagram-platform/instagram-api-with-instagram-login (дата звернення: 06.05.2024)
* PostgreSQL Documentation. URL: https://www.postgresql.org/docs/current/ (дата звернення: 06.05.2024)
* Salunkhe G., Nagpurkar S., Kengale J. A. Boosting Productivity through Deep Learning: Strategies for Enhanced Efficiency. International Journal of Intelligent Systems and Applications in Engineering, 2024, 12(13), 396–406.
* Similarweb PRO. Рейтинг сайтів в Україні. URL: https://www.similarweb.com/ (дата звернення: 06.05.2024)
* Google Maps. URL: https://www.google.com.ua/maps (дата звернення: 06.05.2024)
* Методичні вказівки до лабораторних робіт з дисципліни «Емпіричні методи програмної інженерії» для студентів спеціальності 121 – Інженерія програмного забезпечення / упоряд.: І. В. Груздо, О. С. Назаров. – Харків: ХНУРЕ, 2024. 127 с.
* Методичні вказівки до практичних робіт з дисципліни «Емпіричні методи програмної інженерії» для студентів спеціальності 121 – Інженерія програмного забезпечення / упоряд.: І. В. Груздо, О. С. Назаров. – Харків: ХНУРЕ, 2021. 80 с.
* Методичні вказівки до лабораторних робіт з дисципліни «Юзабіліті» для студентів спеціальності 121 – Інженерія програмного забезпечення / упоряд.: І. В. Груздо, Д. О. Колесников, А. О. Функендорф. – Харків: ХНУРЕ, 2020. 76 с.
* Системні технології, № 4(153) 2024. “System technologies”, 47–57. DOI: 10.34185/1562-9945-4-153-2024-06
* Пироженко С. С., Лєсна Н. С. Дослідження продуктивності методів та засобів контролю глобального стану компонентів React‐додатків. Наука онлайн: міжнародний електронний науковий журнал, 2020, № 10, 13 с. URL: https://nauka-online.com/.../doslidzhennya-produktivnosti-... (дата звернення: 06.05.2024)
* Соловей І. В. Розробка програмної системи для ведення цифрової картки домашньої тварини. Системні технології, 2024, № 4(153), 47–57. DOI: 10.34185/1562-9945-4-153-2024-06
* Методичні вказівки до курсового проектування з дисципліни «Бази даних» для студентів спеціальності 121 – Інженерія програмного забезпечення / упоряд.: О. О. Мазурова та ін. – Харків: ХНУРЕ, 2020. 52 с. URL: https://catalogue.nure.ua/download=237424 (дата звернення: 06.05.2024)

# Overall Description

## Product Perspective

Веб-застосунок «Сервіс для вирішення питань, пов’язаних із замовленнями в піцерії» є автономним продуктом, який не входить до існуючої продуктової лінійки, але призначений для інтеграції з платіжними та логістичними сервісами піцерії. Система складається з двох основних компонентів:

* Бекенд (обробка даних, бізнес-логіка, взаємодія з базою даних і зовнішніми API);
* Фронтенд (інтерфейс користувача для веб та мобільних пристроїв).
* Взаємодія між клієнтом і сервером відбувається через REST API, що дозволяє підключати сторонні сервіси (наприклад, служби доставки).

## Product Features

Основні функції системи включають:

* авторизація та реєстрація: email, Google OAuth;
* каталог піц: детальна інформація, сортування, фільтрація збережених параметрів у URI;
* кастомізація піци: вибір розміру, інгредієнтів;
* кошик: додавання, редагування кількості, видалення позицій;
* оформлення замовлення: введення адреси доставки та контактних даних;
* оплата: інтеграція з LiqPay;
* сповіщення: email-лісти про статус замовлення;
* маркетингові модулі: публікація акцій через Instagram Stories;
* адаптивний дизайн: підтримка десктопів, планшетів.

## User Classes and Characteristics

* Гість (Guest): перегляд каталогу, перегляд Instagram Stories, створеня власної піци.
* Зареєстрований користувач (User): перегляд каталогу, перегляд Instagram Stories, створеня власної піци, створення замовлень, оплата, сповіщення.
* Менеджер піцерії (Manager): перегляд і обробка замовлень.
* Адміністратор (Admin): повний доступ, налаштування системи, аналітика.

## Operating Environment

Система працює в клієнт-серверному середовищі:

* Сервер: Node.js/Next.js на Vercel, база даних PostgreSQL (Neon).
* Клієнт: сучасні веб-браузери (Chrome, Firefox, Safari, Edge) та мобільні WebView.
* З’єднання: HTTPS для всіх API-запитів.

## Design and Implementation Constraints

* Технології: Next.js 15, React 19, Tailwind CSS;
* бекенд: безсерверні Edge-функції на Vercel;
* ORM: Prisma для роботи з PostgreSQL;
* авторизація: NextAuth (email-magic link, Google OAuth 2.0);
* стейт-менеджмент: Zustand;
* email-сервіс: Resend;
* безпека: SSL/TLS, шифрування паролів bcryptjs, валідація Zod.

## User Documentation

* Вбудована довідка: підказки інтерфейсу (tooltips, модальні вікна);
* API-документація: опис REST-ендпоінтів та приклади запитів/відповідей.

## Assumptions and Dependencies

Припущення:

* користувачі мають стабільне інтернет-з’єднання;
* користувачі мають базові навички роботи з веб-інтерфейсами.

Залежності:

* платформа розгортання Vercel;
* зовнішні сервіси: LiqPay, Google Maps API (для вибору адреси), Instagram Graph API.

# System Features

**3.1 User Registration and Authentication**

3.1.1 Description and Priority

Реєстрація та авторизація користувачів із підтримкою email-посилання та Google OAuth. Пріоритет: високий (необхідна для оформлення замовлень).

Створення своєї піци. Пріоритет: високий.

3.1.2 Stimulus/Response Sequences

Стимул: незареєстрований користувач відкриває сторінку реєстрації та заповнює форму (ПІБ, email, пароль) для оформлення замовленя.

Відповідь: система перевіряє, що email ще не зареєстрований, хешує пароль (BCrypt), створює профіль і надсилає повідомлення про успішну реєстрацію.

Стимул: вже зареєстрований користувач намагається зареєструватися з тим самим email.

Відповідь: система повертає повідомлення про те, що такий email уже використовується, і пропонує перейти на сторінку входу.

Стимул: користувач відкриває сторінку входу і вводить свій email і пароль.

Відповідь: система перевіряє дані, видає JWT-токен сесії та перенаправляє в особистий кабінет.

Стимул: користувач натискає “Увійти через Google”.

Відповідь: система перенаправляє до Google OAuth, при успіху перевіряє наявність профілю за providerId, створює новий, якщо необхідно, та видає токен сесії.

Стимул: користувач вводить неправильний email/пароль.

Відповідь: система повертає коректне повідомлення про помилку (неправильні дані).

Стимул: неавторизований користувач хоче замовити товар і оплатити його.

Відповідь: система повертає коректне повідомлення про помилку і перенаправляє на сторінку /not-auth.

3.1.3 Functional Requirements

REQ-1: підтримка реєстрації та входу через email/password.

REQ-2: підтримка реєстрації та входу через Google OAuth.

REQ-4: хешування паролів bcrypt.

REQ-5: строки сесії мають закінчуватися через 24 години неактивності.

REQ-6: здійсненя оплати через LiqPay.

REQ-6: захист від SQL-ін’єкцій і XSS-атак через валідацію введених даних за допомогою Prisma.

**3.2 User Profile Management**

3.2.1 Description and Priority

Управління профілями користувачів із можливістю редагування контактних даних і паролю. Пріоритет: середній.

3.2.2 Stimulus/Response Sequences

Стимул: користувач відкриває налаштування профілю.

Відповідь: система відображає поточні дані (ім’я, email, та прихований пароль).

Стимул: користувач відкриває налаштування профілю та хоче змінити пароль.  
Відповідь: система відображає форму з трьома полями: «поточний пароль», «новий пароль», «підтвердження нового пароля».

Стимул: користувач оновлює адресу доставки.

Відповідь: система перевіряє формат адреси через Google Maps API і зберігає нову адресу.

3.2.3 Functional Requirements

REQ-6: профіль містить поля: ПІБ, email, фото, кафедра, група, курс, бали, місце в рейтингу, приватність (isPrivate).

REQ-7: можливість редагування профілю (крім email) і налаштування приватності. назва).

REQ-8: зміна пароля через форму: поточний пароль + новий пароль + підтвердження;

REQ-9: перевірка відповідності поточного пароля, збіг нових полів та мінімальні вимоги до складності нового пароля (8+ символів, букви та цифри);

REQ-10: хешування нового пароля bcrypt і заміна старого.

**3.3 Order Processing**

3.3.1 Description and Priority

Моніторинг і управління академічними досягненнями (оцінки, нагороди, сертифікати). Пріоритет: високий.

3.3.2 Stimulus/Response Sequences

Стимул: користувач додає піцу в кошик.

Відповідь: система створює або оновлює позицію кошика з коректною кількістю.

Стимул: користувач оформлює замовлення.

Відповідь: система створює запис замовлення з деталями (страви, адреса, сума).

Стимул: користувач переглядає статус замовлення.

Відповідь: система повертає поточний статус (нове, в обробці, доставляється, завершено).

3.3.3 Functional Requirements

REQ-11: кожне замовлення містить ID, список страв, суму, адресу та статус.

REQ-12: можливість змінювати кількість і видаляти позиції до оплати.

REQ-13: автоматичний розрахунок загальної суми при кожній зміні кошика.

**3.4 Payment Processing**

3.4.1 Description and Priority

Інтеграція з платіжними сервісами для безпечної онлайн-оплати замовлень. Пріоритет: високий.

3.4.2 Stimulus/Response Sequences

Стимул: користувач натискає «Оплатити замовлення».

Відповідь: система формує запит до LiqPay API, отримує платіжну сторінку і перенаправляє користувача.

Стимул: LiqPay повідомляє про успішну оплату через webhook.

Відповідь: система оновлює статус замовлення на «Оплачено» та надсилає підтвердження email.

3.4.3 Functional Requirements

REQ-14: інтеграція з LiqPay (REST API) для створення інвойсу та перевірки статусу платежу.

REQ-15: безпечний обмін даними через HTTPS і підпис SHA-1.

REQ-16: оновлення статусу замовлення після отримання callback від LiqPay.

REQ-17: надсилання email-підтвердження про успішну оплату.

**3.5 Cart Management**

3.5.1 Description and Priority

Модуль керування кошиком: зміна кількості та видалення позицій. Пріоритет: високий.

3.5.2 Stimulus/Response Sequences

Стимул: користувач натискає «Оплатити замовлення».

Відповідь: система формує запит до LiqPay API, отримує платіжну сторінку і перенаправляє користувача.

Стимул: LiqPay повідомляє про успішну оплату через webhook.

Відповідь: система оновлює статус замовлення на «Оплачено» та надсилає підтвердження email.

3.5.3 Functional Requirements

REQ-18: інтеграція з LiqPay (REST API) для створення інвойсу та перевірки статусу платежу.

REQ-19: безпечний обмін даними через HTTPS і підпис SHA-1.

REQ-20: оновлення статусу замовлення після отримання callback від LiqPay.

REQ-21: надсилання email-підтвердження про успішну оплату.

**3.6 Cart Management**

3.6.1 Description and Priority

Модуль керування кошиком: зміна кількості та видалення позицій. Пріоритет: високий.

3.6.2 Stimulus/Response Sequences

Стимул: користувач збільшує/зменшує кількість піцi в кошику.

Відповідь: система отримує PATCH-запит, оновлює quantity, перераховує totalAmount і повертає оновлений кошик.

Стимул: користувач видаляє позицію з кошика.

Відповідь: система отримує DELETE-запит, видаляє CartItem і повертає новий totalAmount.

3.6.3 Functional Requirements

REQ-22: підтримка PATCH /api/cart/:itemId для оновлення quantity.

REQ-23: підтримка DELETE /api/cart/:itemId для видалення позиції.

REQ-24: при кожній зміні автоматичний перерахунок totalAmount.

REQ-25: захист від неавторизованих запитів через перевірку JWT

**3.7 Product Filtering and Recommendations**

3.7.1 Description and Priority

Фільтрація каталогу за ціною, розміром піци та типом продукту; персоналізовані рекомендації. Пріоритет: середній.

3.7.2 Stimulus/Response Sequences

Стимул: користувач встановлює фільтр за ціною (від–до) або розміром (маленька, середня, велика).

Відповідь: система формує SQL-запит із відповідними умовами та повертає відфільтрований список товарів.

Стимул: користувач вибирає категорії (піца, кава, нагетси тощо).

Відповідь: система фільтрує за productType і повертає релевантні позиції.

Стимул: користувач переглядає сторінку конкретного товару.

Відповідь: система аналізує історію переглядів та пропонує блок «Рекомендовані товари» на основі популярності та схожості.

3.7.3 Functional Requirements

REQ-26: підтримка фільтрації за ціною та розміром через query-параметри (minPrice, maxPrice, size).

REQ-27: фільтрація за productType (pizza, coffee, nuggets, тощо).

REQ-28: пагінація результатів фільтрації.

REQ-29: алгоритм рекомендацій: на основі сумісних товарів.

REQ-30: відображення блоку «Рекомендовані товари» на сторінці продукту.

REQ-31: відображення модального вікна для вибору інгрідієнтів.

# External Interface Requirements

## User Interfaces

* Сторінка реєстрації/входу: поля для email, пароля, кнопка «Увійти через Google», валідація (bcrypt).
* Каталог товарів: список продуктів (піца, кава і т.п), фільтри за ціною, розміром, типом продукту; сортування; пагінація.
* Сторінка товару: деталі (опис, інгредієнти, ціни), вибір розміру, додавання до кошика, блок «Рекомендовані товари».
* Кошик: динамічний список позицій із кнопками +/– для зміни кількості та кнопкою видалення; відображення загальної суми.
* Сторінка оформлення замовлення: поля для адреси (Autocomplete Google Maps), контактів, вибір способу оплати; кнопка «Оплатити».
* Сторінка акцій: перелік активних промо-кампаній з банерами, описами та термінами дії.
* Insta stories у головному меню.

## Hardware Interfaces

* Система працює на стандартних серверах із підтримкою JVM для Spring Boot.
* Клієнтські пристрої: десктопи, ноутбуки, планшети, смартфони з веб-браузерами.
* Немає специфічних вимог до апаратного забезпечення, крім стабільного інтернет-з’єднання.
* Сервер: безсерверні Edge-функції на Vercel або Node.js-сервер; база даних PostgreSQL (Neon) або інша хмарна СУБД.
* Клієнти: десктопи, ноутбуки, планшети,з будь-яким сучасним браузером.

## Software Interfaces

* Prisma ORM для взаємодії з PostgreSQL.
* NextAuth.js для аутентифікації (Google OAuth, email).
* @react-google-maps/api + react-google-autocomplete для вибору адреси.
* LiqPay REST API для створення та перевірки платежів; webhook-роут для обробки callback.
* zustand для управління станом кошика.
* react-hot-toast для сповіщень.
* TailwindCSS для стилізації, shadcn/ui для компонентів.

## Communications Interfaces

* HTTPS для всіх запитів.
* REST API (GET, POST, PATCH, DELETE) із JSON.
* LiqPay webhook для підтвердження платежів.
* Email-сервіс (Resend) для сповіщень клієнтам.

# Other Nonfunctional Requirements

## Performance Requirements

* Час загрузки сторінки каталогу: ≤ 3 с (LCP ≤ 2.5 с).
* Обробка до 300 одночасних користувачів без деградації.
* Створення інвойсу та перенаправлення на сторінку платежу: ≤ 2 с.

## Safety Requirements

* Механізм rollback транзакцій при помилках імпорту меню чи замовлень.
* Відсутність єдиної точки відмови — горизонтальне масштабування Edge-функцій.

## Security Requirements

* Аудит дій користувачів для відстеження несанкціонованого доступу.
* Хешування паролів bcrypt.
* Захист від XSS, CSRF (Next.js CSRF middleware), SQL-ін’єкцій (Prisma ORM).
* JWT-сесії.

## Software Quality Attributes

* Масштабованість: підтримка горизонтального та вертикального масштабування.
* Надійність: 99.9% доступності системи.
* Зручність: інтуїтивна навігація, адаптивний дизайн.
* Maintainability: чиста архітектура (багато файлова структура, модульність компонентів).Observability: логування запитів і помилок (Vercel Logs, Sentry).

# Other Requirements

* Локалізація: українська за замовчуванням, англійська в майбутніх версіях.
* Логування: всі дії адміна, зміни кошика й статусу замовлення зберігаються в логах.
* Соціальні мережі: Open Graph-теги для поділу акцій у Facebook/Instagram.
* Моніторинг: інтеграція з UptimeRobot або Datadog для перевірки доступності сервісу.

Appendix A: Glossary

* SSO (Single Sign-On) — механізм єдиного входу, що дозволяє користувачеві автентифікуватися в системі через зовнішні провайдери, наприклад Google OAuth.
* REST API — інтерфейс для обміну даними між клієнтом і сервером за допомогою HTTP-запитів із тілом у форматі JSON.
* JWT (JSON Web Token) — компактний токен у форматі JSON для передачі закодованої інформації про сесію користувача.
* Next.js — React-фреймворк для побудови серверно-рендерених і статично згенерованих веб-додатків.
* Prisma ORM — інструмент об’єктно-реляційного відображення для взаємодії з базою даних PostgreSQL та іншими СУБД.
* LiqPay — платіжна платформа з REST-інтерфейсом для створення інвойсів та обробки онлайн-платежів.
* bcrypt — алгоритм хешування паролів із налаштовуваною складністю (salt rounds) для безпечного зберігання.
* CDN (Content Delivery Network) — мережа серверів для кешування й швидкої доставки статичних ресурсів.
* Webhook — механізм push-оповіщення від зовнішніх сервісів на ваш API-ендпоінт.
* CSRF (Cross-Site Request Forgery) — захист від міжсайтових підробок запитів, що гарантує, що дії ініціюються лише довіреним джерелом.
* XSS (Cross-Site Scripting) — захист від виконання шкідливого JavaScript-коду, який може бути вставлений користувачем.
* HTTPS — протокол захищеної передачі даних поверх SSL/TLS, що гарантує конфіденційність і цілісність трафіку.

Appendix B: Analysis Models

Діаграма класів: описує основні сутності та зв’язки між ними в розроблюваному програмному забезпеченні для веб-застосунку піцерії. (рисунок B.1).

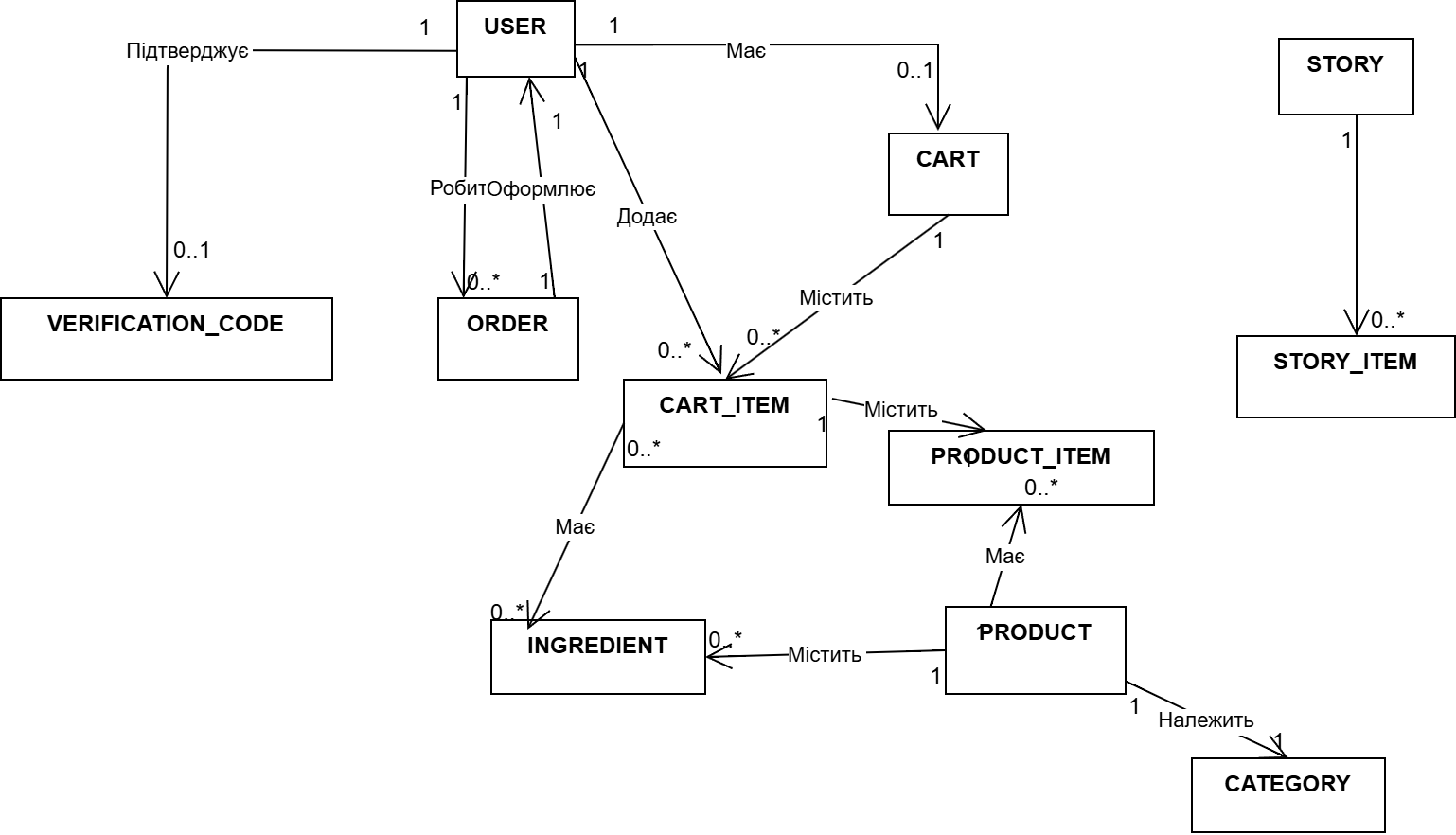


Рисунок B.1 – Діаграма класів

Для зберігання даних у системі було спроектовано наступні основні сутності та їх атрибути, які детально представлені на ER-діаграмі (рисунок B.2).

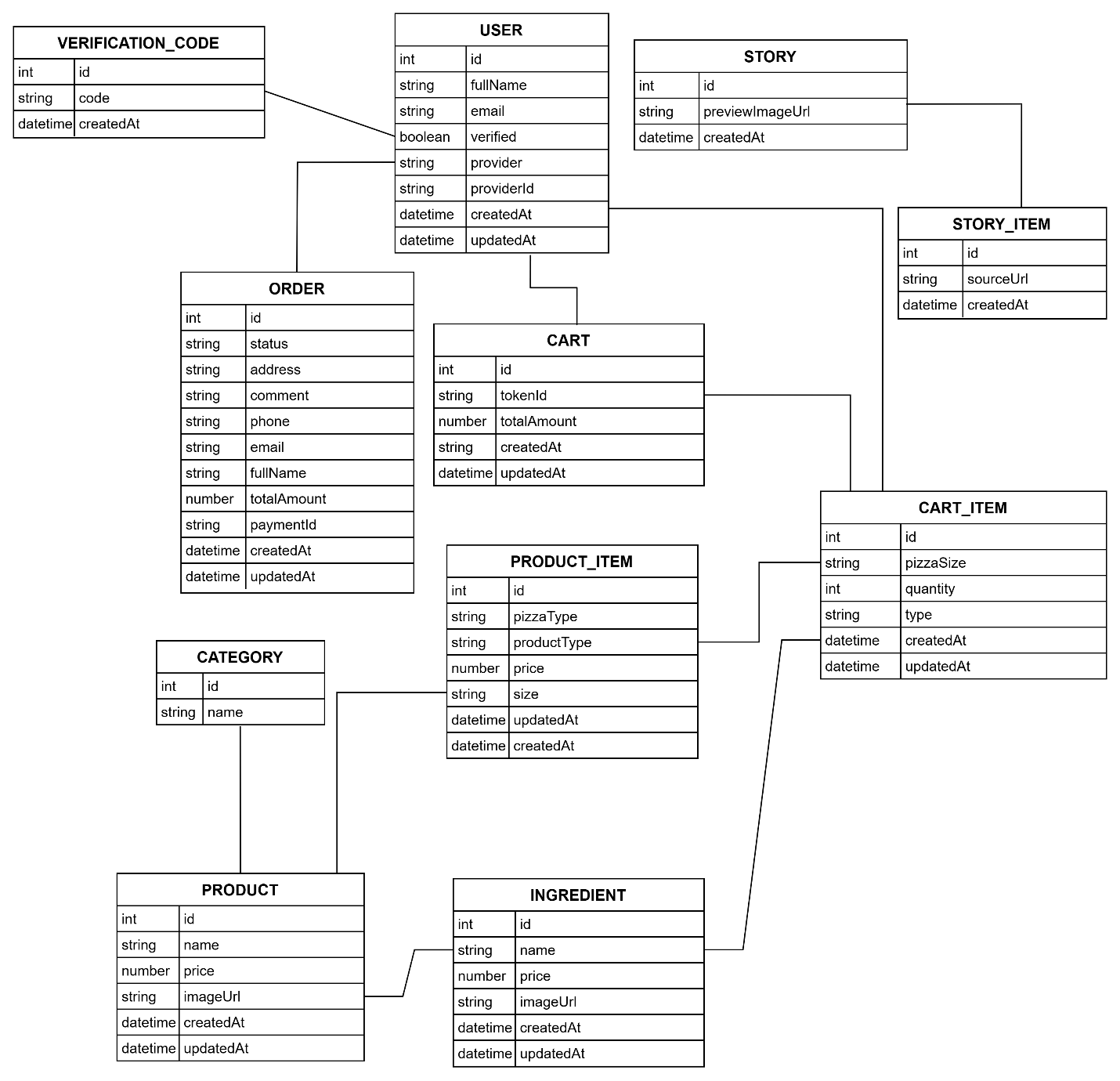


Рисунок B.2 – ER-діаграма для сутностей

Було створено діаграму розгортання для веб-застосунку піцерії. Діаграма розгортання дає змогу відобразити фізичну структуру системи, взаємодію між її компонентами, а також їхнє розміщення на апаратних вузлах. (рисунок B.3).

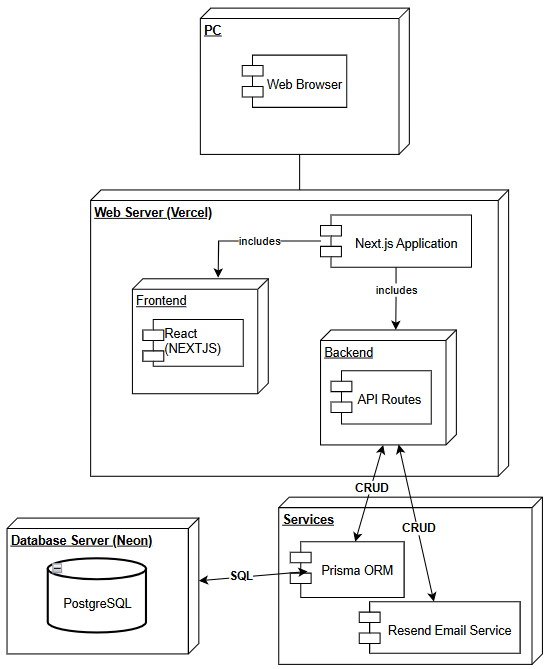


Рисунок B.3 – UML діаграма розгортання веб-застосунку піцерії

Було зображено діаграму потоків даних (DFD) веб-застосунку, що показує процеси обміну інформацією між користувачем, внутрішніми процесами системи та сховищами даних. (рисунок B.4).

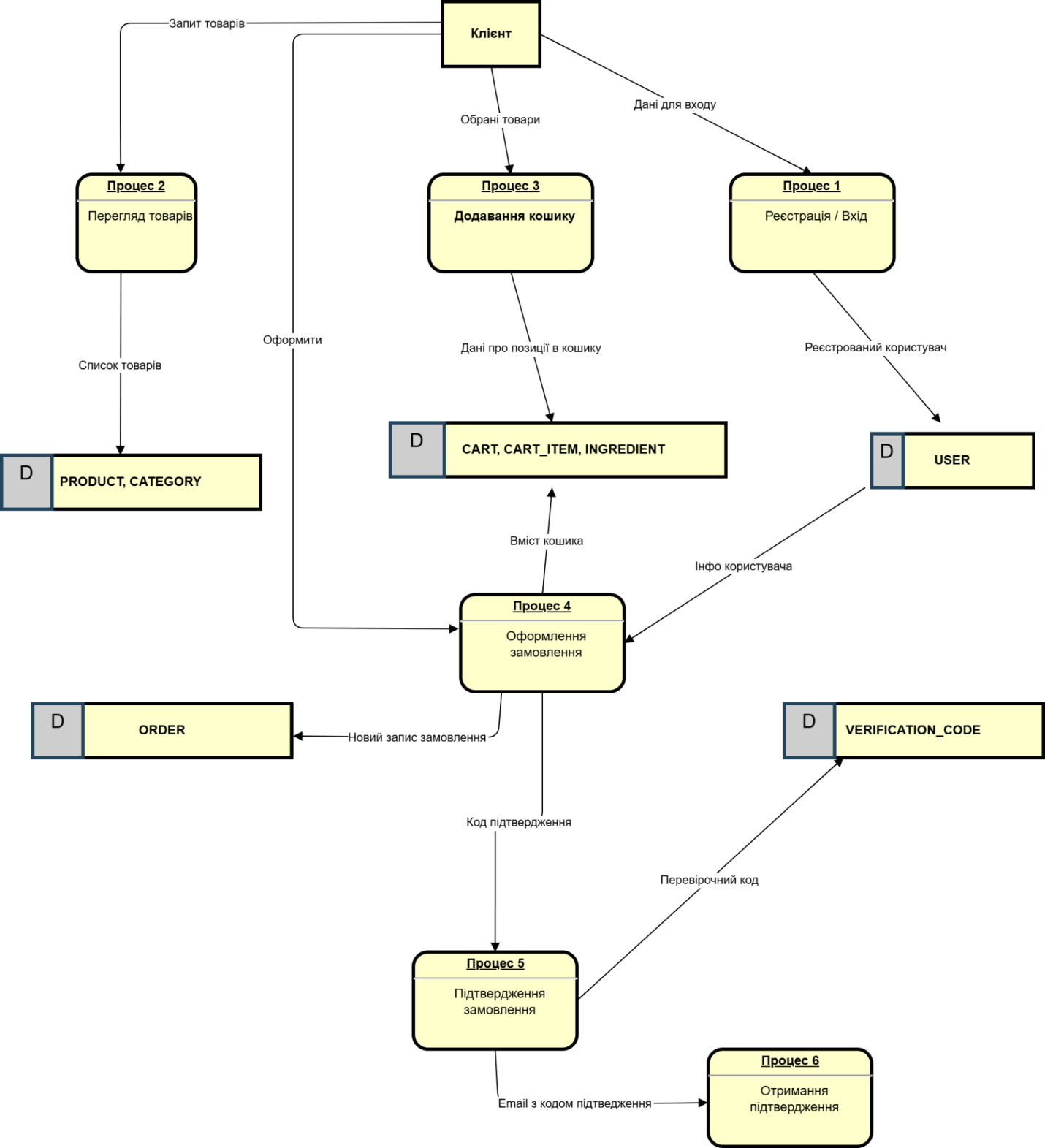


Рисунок B.4 – Діаграма потоків

Appendix C: Issues List

* TBD-2: налаштування резервного каналу платежів на випадок відмови LiqPay
* TBD-3: оптимізація кешування та CDN для швидкої доставки меню при великому навантаженні
* TBD-4: вдосконалення алгоритму рекомендацій на основі поведінки користувачів
* TBD-5: додавання багатомовності інтерфейсу (укр., англ.)
* TBD-6: розробка мобільної версії або PWA для зручності замовлення з телефону
* TBD-7: інтеграція з системою аналітики для моніторингу конверсії та популярних позицій меню