

РОЗРОБКА WEB-ДОДАТКУ «ІНТЕРНЕТ МАГАЗИН ПОБУТОВИХ ТОВАРІВ»

¹Вінницький національний технічний університет

Анотація

У доповіді викладено результати розробки WEB-додатку для продажу побутових товарів, призначеного для автоматизації продажів у малих та середніх підприємствах. Запропоновано багатoshарову архітектуру на базі Spring Boot і Thymeleaf, спроектовано реляційну базу даних у 3НФ із сутностями users, products, carts, orders тощо. Описано UML-моделі (прецедентів, класів, послідовності) і алгоритми основних процесів: реєстрації, формування кошика, оформлення замовлення. Проведено первинне тестування, яке підтвердило коректність бізнес-логіки.

Ключові слова: WEB-додаток, Spring Boot, реляційна база даних, UML, електронна комерція.

Abstract

This paper presents the development of a web application for selling household goods, designed to automate sales for small and medium enterprises. A multi-layer architecture based on Spring Boot and Thymeleaf is proposed, and a third-normal-form relational database with entities users, products, carts, orders, etc., is designed. UML models (use-case, class, sequence) and algorithms for core processes—registration, cart management, order placement—are described. Preliminary testing confirmed the correctness of business logic.

Keywords: WEB application, Spring Boot, relational database, UML, e-commerce.

Вступ

У сучасних умовах стрімкого розвитку електронної комерції все більше споживачів віддають перевагу онлайн-покупкам побутових товарів — від засобів гігієни та побутової хімії до дрібної кухонної техніки й аксесуарів для дому. Зручність, економія часу та можливість порівняти десятки пропозицій одним кліком стимулюють подальший розвиток цієї галузі. Водночас масові маркетплейси часто пропонують надто широкий асортимент, перевантажений інтерфейс, фрагментарну підтримку клієнтів та відсутність чіткої спеціалізації саме на побутових товарах.

Багато платформ-лідерів мають слабкі місця, що заважають користувачам швидко і комфортно здійснювати замовлення. Серед поширених недоліків — заплутані меню й фільтри, нерозвинена програма лояльності, обмежена можливість персоналізації профілю та неповна прозорість інформації про товар. Окрім того, власники бізнесу змушені вкладати значні ресурси в оренду складів, ручне адміністрування асортименту та роботу служби підтримки.

Для оптимізації процесу продажу побутових товарів актуальним є створення спеціалізованого WEB-додатка з вузькою фокусованістю. Такий застосунок поєднає інтуїтивно зрозумілий інтерфейс, гнучкі фільтри пошуку, прозоре відображення характеристик, а також ефективні інструменти адміністрування з чітким розподілом ролей (Admin / Client). Інтеграція сучасного технологічного стеку забезпечить масштабованість, безпеку й простоту подальшого розширення функціоналу.

Метою дослідження є розробка WEB-додатка для продажу побутових товарів із спеціалізованим інтерфейсом, ефективною системою фільтрації та персоналізації, а також повноцінною адміністративною панеллю.

Результати дослідження

У результаті проведеного дослідження було створено детальний технічний проєкт і реалізовано функціональний прототип WEB-додатку «Інтернет-магазин побутових товарів», що відповідає сучасним вимогам електронної комерції малого й середнього бізнесу. На етапі аналізу предметної області було вивчено тенденції розвитку онлайн-торгівлі, виявлено ключові обмеження існуючих маркетплейсів та сформульовано набір бізнес-вимог, які покликані підвищити юзабіліті та ефективність процесу купівлі. Ці вимоги стало можливим відобразити в наборі чітко визначених сценаріїв використання (Use Case), які лягли в основу подальшої архітектури системи.

Після формування функціональних та нефункціональних вимог було обґрунтовано вибір трисегментної архітектури клієнт–сервер–дані. Клієнтська частина, реалізована за допомогою серверного шаблонізатора Thymeleaf [1] у поєднанні з фреймворком Bootstrap, забезпечує адаптивність інтерфейсу та швидкий час відгуку на різних пристроях. Серверна логіка сформована на основі Spring Boot[2] із використанням Spring Security[3] для аутентифікації та авторизації, а також Spring Data JPA[4] для зручної роботи з реляційною базою даних. Такий підхід дозволив відокремити відповідальність за представлення, бізнес-логіку та зберігання даних, спростивши тестування й подальше масштабування проєкту.

Особливу увагу приділено проєктуванню бази даних. На основі аналізу предметної області розроблено ER–модель, яка включає сутності: users, categories, manufacturers, products, carts, cart_items, orders та order_items. Застосування третьої нормальної форми (3НФ) до структури таблиць гарантувало відсутність надлишковості даних і мінімізацію аномалій при вставці, оновленні та видаленні записів. Було визначено оптимальні індекси для прискорення пошуку за ключовими полями, а також налаштовано механізм ініціалізації початкових даних за допомогою SQL-скриптів.

Для візуалізації внутрішньої структури та взаємодії компонентів використано набір UML-діаграм [5]. Діаграма класів дала змогу описати основні сутності й їх атрибути, а також показати зв'язки та залежності між ними. Діаграми послідовності демонструють детальний порядок викликів у процесах додавання товару в кошик і оформлення замовлення, що важливо для коректної реалізації контролерів REST API[6]. Діаграма розгортання ілюструє фізичне розміщення компонентів на вузлах: браузер клієнта, веб-сервер та сервер бази даних, що дає змогу планувати інфраструктуру й розподіляти навантаження між сервісами.

У процесі розробки було реалізовано базові бізнес-функції, серед яких: реєстрація та авторизація користувачів із захистом від CSRF і хешуванням паролів через BCrypt; керування особистим кабінетом з можливістю редагування контактних і платіжних даних; гнучка система пошуку й фільтрації товарів за категоріями, брендами, ціною й рейтингом; формування кошика з можливістю коригування кількості позицій; та функціонал оформлення замовлення із вибором способів доставки й оплати. Адміністративний інтерфейс дозволяє виконувати CRUD-операції над товарами, категоріями, виробниками та переглядати статистику замовлень.

Тестування прототипу включало модульні тести сервісів із використанням JUnit [7] і Mockito[8], а також інтеграційні тести REST-контролерів. Час відповіді API на середньому навантаженні (до 500 одночасних запитів) не перевищував 300–400 мс, що відповідає заданим нефункціональним вимогам. Ручне тестування клієнтського інтерфейсу підтвердило коректне відображення сторінок на десктопах та мобільних пристроях, а також зручність користування.

Фінальний результат дослідження — готовий до промислового використання прототип, що повністю задовольняє визначені цілі та завдання. Структура проєкту побудована таким чином, щоб забезпечувати просту інтеграцію з платіжними системами та зовнішніми логістичними сервісами, а також можливість подальшого розширення функціоналу – наприклад, додавання модулів аналітики, програми лояльності, інтеграції з мобільним додатком.

Таким чином, результати дослідження свідчать про успішне виконання всіх етапів розробки: від збору та аналізу вимог, проєктування архітектури й бази даних, моделювання UML-діаграм і опису алгоритмів, до реалізації, тестування та верифікації ключових бізнес-процесів. Розроблений WEB-додаток готовий до впровадження та подальшого використання в рамках малого й середнього підприємництва.

Висновки

Розроблено повнофункціональний прототип WEB-додатку «Інтернет-магазин побутових товарів» із багаторівневою архітектурою на Spring Boot і Thymeleaf, нормалізованою реляційною базою даних у

ЗНФ та чітко задокументованими UML-діаграмами. Реалізовано ключові бізнес-процеси — реєстрацію, пошук і фільтрацію товарів, роботу з кошиком і оформлення замовлення — із захистом даних і адаптивним інтерфейсом. Проведене тестування підтвердило відповідність системи вимогам щодо продуктивності та надійності. Розроблений додаток готовий до впровадження та подальшого масштабування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Thymeleaf. Thymeleaf: Java XML/XHTML/HTML5 template engine [Електронний ресурс] // Режим доступу: <https://www.thymeleaf.org> ; Дата звернення: 21.05.2025.
2. Pivotal Software, Inc. Spring Boot Reference Documentation [Електронний ресурс] // Режим доступу: <https://docs.spring.io/spring-boot/index.html> ; Дата звернення: 21.05.2025.
3. Pivotal Software, Inc. Spring Security [Електронний ресурс] // Режим доступу: <https://spring.io/projects/spring-security> ; Дата звернення: 21.05.2025.
4. Spring Data JPA [Електронний ресурс] // Режим доступу: <https://spring.io/projects/spring-data-jpa> ; Дата звернення: 29.05.2025.
5. Каграманова Ю. Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти [Електронний ресурс] // Режим доступу: <https://dou.ua/forums/topic/40575/> ; Дата звернення: 21.05.2025.
6. Liew Z. Як правильно працювати з REST API [Електронний ресурс] // Режим доступу: <https://itvdm.com/ua/blog/article/rest-api-18> ; Дата звернення: 21.05.2025.
7. JUnit 5 [Електронний ресурс] // Режим доступу: <https://junit.org/junit5/> ; Дата звернення: 29.05.2025.
8. Mockito [Електронний ресурс] // Режим доступу: <https://site.mockito.org> ; Дата звернення: 29.05.2025.

Нілова Марія Геннадіївна – студентка групи 4КН-21б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, email: mariaaamil.2231@gmail.com.

Сілагін Олексій Віталійович - канд. техн. наук, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, e-mail: avsilagin@vntu.edu.ua.

Nilova Mariia Hennadiivna – student of Group 4KN-21b, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, Ukraine; e-mail: mariaaamil.2231@gmail.com.

Silagin Oleksii Vitaliiiovych – PhD in Technical Sciences, Associate Professor of the Department of Computer Science, Vinnytsia National Technical University, Vinnytsia, Ukraine; e-mail: avsilagin@vntu.edu.ua.