UDK 004.4:658.7

Khoshaba O.M.

Fokin O.I.

DESIGN AND DEVELOPMENT OF AN ORDER AND INVENTORY MANAGEMENT SYSTEM FOR SMALL BUSINESSES

Vinnitsia National Technical University

Анотація.

Малі підприємства часто покладаються на електронні таблиці та розрізнені інструменти для управління замовленнями клієнтів, поповненням запасів та складськими запасами, що призводить до невідповідних записів про запаси, дефіциту запасів, якого можна уникнути, та затримок виконання замовлень. У тезах представлено концепцію та специфікацію легкого веб-програмного інструменту, який інтегрує управління замовленнями з контролем запасів для малих підприємств. Рішення структуровано навколо чітко визначених обмежених контекстів (продажі, запаси, закупівлі, звітність) та застосовує доменно-орієнтований дизайн для моделювання бізнес-сутностей та інваріантів. Запропоновано RESTful HTTP-інтерфейс для підтримки сумісності та майбутніх інтеграцій, тоді як шаблони корпоративних додатків використовуються для роз'єднання інтерфейсу користувача, служб додатків та рівня персистентності. Точність інвентаризації за одночасних операцій забезпечується за допомогою незмінного журналу руху запасів, що дозволяє проводити реконструкцію балансу та забезпечувати узгодженість транзакцій. Оскільки емпіричні результати ще недоступні, у тезах визначено план оцінки, що включає автоматизовану перевірку основних інваріантів запасів, тестування зручності використання на основі завдань з репрезентативними операторами та оцінку продуктивності за допомогою сценаріїв навантаження та показників процентильної затримки. Запропонований підхід має на меті забезпечити прагматичну, економічно ефективну основу для цифровізації виконання замовлень та складських операцій у малих підприємствах.

Ключові слова: *управління замовленнями, управління запасами, управління складом, інформаційні системи для малого бізнесу, проєктування на основі предметної області, RESTful API, реєстр руху запасів, оцінка продуктивності.*

Abstract.

Small businesses frequently rely on spreadsheets and disconnected tools to manage customer orders, replenishment, and warehouse stock, resulting in inconsistent stock records, avoidable stock-outs, and delayed fulfilment. This paper presents the concept and specification of a lightweight web-based software tool that integrates order management with inventory control for small enterprises. The solution is structured around clearly defined

bounded contexts (Sales, Inventory, Procurement, Reporting) and applies domain-driven design to model business entities and invariants. A RESTful HTTP interface is proposed to support interoperability and future integrations, while enterprise application patterns are used to decouple the user interface, application services, and persistence layer. Inventory accuracy under concurrent operations is ensured by an immutable stock-movement ledger, which enables auditable balance reconstruction and transactional consistency. As empirical results are not yet available, the paper defines an evaluation plan comprising automated verification of core inventory invariants, task-based usability testing with representative operators, and performance assessment using load scenarios and percentile latency metrics. The proposed approach is intended to provide a pragmatic, cost-effective foundation for digitising order fulfilment and warehouse operations in small businesses.

Keywords: *order management, inventory control, warehouse management, small business information systems, domain-driven design, RESTful API, stock-movement ledger, performance evaluation.*

Small businesses frequently coordinate customer orders, purchasing, and stockkeeping using spreadsheets and fragmented tools, which increases fulfilment lead time, reduces stock record accuracy, and ties up working capital in avoidable overstock. Although comprehensive ERP suites exist, their acquisition cost, configuration effort, and operational complexity are often disproportionate for small organisations.

This paper proposes a development concept for a lightweight software tool for integrated order and warehouse inventory management, aimed at improving stock visibility, reducing stock-outs and overstocks, and supporting consistent replenishment decisions.

The object of the study is the operational processes of order fulfilment and inventory control in a small-business environment; the subject is the set of information models, algorithms, and architectural decisions required to implement those processes in software.

The functional core is derived from established inventory-management approaches, including ABC classification, economic order quantity, and continuous-review reorder point policies with safety stock [1].

To preserve semantic alignment between business terminology and the software model, the domain is decomposed into bounded contexts (Sales, Inventory, Procurement, Reporting) in line with domain-driven design, enabling explicit modelling of entities, value objects, aggregates, and invariants [2]. The boundaries are selected to support an initial modular-monolith deployment while remaining suitable for subsequent extraction into microservices where justified, including the use of sagas for long-running order and procurement workflows [3]. The system is specified as a web-based client–server application that exposes a uniform, resource-oriented HTTP interface, consistent with REST constraints, to improve evolvability and integration readiness [4]. Enterprise application patterns (service layer, repository, unit of work, and data mapper) are applied to separate presentation, application, and persistence concerns and to reduce coupling to storage technology [5]. Stock accuracy under concurrent updates is treated as a first-class requirement; therefore, all changes (receipts, reservations, picks, adjustments, returns) are modelled as immutable stock-movement events and committed within database transactions to preserve atomicity and isolation of inventory balances [6].

The implementation concept assumes a relational data model centred on Products/SKUs, Locations, Orders, Order Lines, Suppliers/Customers, and Stock Movements, supplemented by role-based access control, audit logging, and configurable business rules (e.g., partial shipments, backorders, minimum order quantities, reorder thresholds). Non-functional requirements are mapped to the ISO/IEC 25010 quality model, prioritising reliability, maintainability, usability, and security for non-technical users [7]. Security controls are defined to address prevalent web-application risks (broken access controls, injection, insecure design, authentication failures) and aligned with the OWASP Top 10 guidance [8]. As empirical results are not currently available, an evaluation methodology is defined: functional correctness through automated

unit/integration tests of critical invariants (e.g., non-negative available-to-promise, conservation of stock across movements), usability assessment via task-based testing with representative operators, and performance assessment via load scenarios (order creation, reservation, picking, reporting) executed with Apache JMeter and analysed using throughput, error rate, and percentile response times [9].

Current limitations include the absence of production datasets and the restricted availability of external integrations (POS, accounting, carrier APIs).

The proposed tool is expected to provide a pragmatic, low-cost foundation for digitising order and inventory operations in small businesses.

Future work will focus on pilot deployment, barcode-assisted warehouse operations, demand forecasting, and multi-warehouse/multi-tenant support.

LIST OF REFERENCES

1. Silver E. A., Pyke D. F., Thomas D. J. Inventory and Production Management in Supply Chains. - 4th ed. - Boca Raton; London; New York: CRC Press, 2017. - 782 p.

2. Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software. - Boston: Addison-Wesley Professional, 2003. - 560 p.

3. Richardson C. Microservices Patterns: With examples in Java. - Shelter Island, NY: Manning Publications, 2018. - 520 p.

4. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures: dissertation. - Irvine, CA: University of California, Irvine, 2000. - 180 p.

5. Fowler M. Patterns of Enterprise Application Architecture. - Boston: Addison-Wesley Professional, 2002. - 560 p.

6. Gray J., Reuter A. Transaction Processing: Concepts and Techniques. - San Mateo, Calif.: Morgan Kaufmann Publishers, 1993. - xxxii, 1070 p.

7. ISO/IEC 25010:2011. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. - Geneva: ISO, 2011. - 34 p.

8. OWASP Foundation. OWASP Top 10:2021 - The Ten Most Critical Web Application Security Risks [Electronic resource]. - 2021. - Available at: https://owasp.org/Top10/ (accessed: 26.01.2026).

9. Apache Software Foundation. Apache JMeter™ User’s Manual [Electronic resource]. - Available at: https://jmeter.apache.org/usermanual/index.html (accessed: 26.01.2026).

Хошаба Олександр Мирославович — канд. техн. наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет

Фокін Олексій Ігорович — студент групи ЗПІ-22б, факультет інформаційних технологій та комп'ютерної інженерії, національний технічний університет, Вінниця, pzmag2022@gmail.com

Khoshaba Oleksandr M. — Cand. Sc. (Eng) Assistant Professor of the Department of Software Engineering, Vinnytsia National Technical University, Vinnytsia

Fokin Oleksiy I. — Department of Software Engineering, Vinnytsia National Technical University, Vinnytsia, email: pzmag2022@gmail.com