UDK 004.4:658.813:658.783

Khoshaba O.M.

Linevych R.U.

ENGINEERING AN ERROR-RESILIENT ORDER AND INVENTORY ACCOUNTING INFORMATION SYSTEM FOR SMALL AND MEDIUM-SIZED ENTERPRISES

Vinnitsia National Technical University

Анотація.

Малі підприємства часто покладаються на електронні таблиці та паперові процеси для відстеження замовлень клієнтів, поставок постачальникам та залишків на складах, що збільшує кількість помилок транскрипції, дублікатів записів та затримок оновлення запасів. У цих тезах представлено концепції проєктування та впровадження веб-інформаційної системи, призначеної для автоматизації обліку замовлень та запасів, одночасно зменшуючи кількість ручних операцій та помилок користувачів. Рішення базується на реляційній моделі бази даних із застосовними обмеженнями цілісності та аудитованим реєстром запасів, побудованим з незмінних записів про рух запасів. Вимоги до якості даних визначені як чіткі бізнес-правила, узгоджені з рамковою системою якості даних ISO 8000 та реалізовані за допомогою комбінації обмежень інтерфейсу користувача, перевірки на стороні сервера та обмежень бази даних. Заходи з перевірки плануються відповідно до методів тестування ISO/IEC/IEEE 29119-4:2021, з акцентом на інваріанти запасів та негативні тести на невдалі перевірки та авторизації. Безпека вирішується за допомогою набору контролю на основі ризиків, що базується на документі Top 10:2021 проєкту Open Worldwide Application Security Project (OWASP), узгодження управління з ISO/IEC 27001:2022 та практик безпечної розробки від Національного інституту стандартів і технологій (NIST) щодо безпечної розробки програмного забезпечення (SSDF). Цілі доступності зіставлені з Керівними принципами доступності веб-контенту (WCAG) 2.2 Консорціуму Всесвітньої павутини (W3C) для підтримки інклюзивної взаємодії з низьким рівнем помилок. Оскільки показники виробництва ще недоступні, у тезах пропонується план оцінки, що заснований на сценарно-орієнтованій функціональній оцінці, показниках якості даних, показниках зручності використання та критеріях ефективності, що підходять для типових робочих навантажень малого бізнесу.

Ключові слова: *малі та середні підприємства, система бухгалтерської інформації, обробка замовлень, облік запасів, управління запасами на складі, якість даних, безпечна розробка програмного забезпечення, веб-додаток, реляційна база даних, тестування програмного забезпечення, доступність.*

Abstract.

Small businesses frequently rely on spreadsheets and paper-based processes to track customer orders, supplier deliveries, and warehouse balances, which increases transcription errors, duplicate records, and delayed stock updates. This paper presents the design and implementation concepts of a web-based information system intended to automate order and inventory accounting while reducing manual operations and user errors. The solution is based on a relational database model with enforceable integrity constraints and an auditable inventory ledger built from immutable stock-movement records. Data quality requirements are defined as explicit business rules, aligned with the ISO 8000 data quality framework, and implemented through a combination of user-interface constraints, server-side validation, and database constraints. Verification activities are planned in accordance with ISO/IEC/IEEE 29119-4:2021 test techniques, with emphasis on inventory invariants and negative tests to validate and authorise failures. Security is addressed through a risk-based control set informed by the Open Worldwide Application Security Project (OWASP) Top 10:2021, governance alignment with ISO/IEC 27001:2022, and secure development practices from the National Institute of Standards and Technology (NIST) Secure Software Development Framework (SSDF). Accessibility objectives are mapped to the World Wide Web Consortium (W3C) Web Content Accessibility Guidelines (WCAG) 2.2 to support inclusive, low-error interaction. As production metrics are not yet available, the paper proposes an evaluation plan based on scenario-driven functional assessment, data-quality indicators, usability measures and performance criteria suitable for typical small-business workloads.

Keywords: *small and medium-sized enterprises, accounting information system, order processing, inventory accounting, warehouse stock control, data quality, secure software development, web application, relational database, software testing, accessibility.*

Small businesses often manage customer orders, supplier deliveries, and warehouse balances using spreadsheets, messaging applications, and paper-based artefacts, which constrain scalability and increase the risk of transcription errors, duplicate records, delayed updates to stock balances, and inconsistent master data. These issues are not merely operational inconveniences; they undermine the accuracy of purchase decisions, inflate safety buffers due to uncertainty, and degrade customer service through avoidable stock-outs and missed delivery commitments.

Recent international analyses emphasise that small and medium-sized enterprises (SMEs) continue to lag behind larger firms in adopting integrated digital tools and structured internal controls, and that the productivity and resilience benefits of digitalisation tend to accrue to organisations that modernise core operational processes rather than adding isolated applications on top of manual practices [1, 2].

In this context, the development of a focused information system for order processing and inventory accounting is a relevant applied task, particularly when engineered to minimise manual operations and user error while preserving traceability, data quality, and security.

This work presents the design and implementation concept of a web-based information system that automates the accounting of orders, supplies, and on-hand inventory for small-business settings characterised by limited information technology staffing, moderate transaction volumes, and a high sensitivity to mistakes during routine data entry.

The practical objective is to reduce manual ledger updates, decrease the incidence of user-induced errors and ensure timely and consistent visibility of stock positions across the order fulfilment lifecycle.

The development approach follows an iterative life cycle that combines requirements elicitation, domain modelling, and incremental prototyping, with an explicit emphasis on data quality governance, security-by-design, accessibility, and systematic verification.

Data quality requirements are formalised as enforceable business rules covering uniqueness, completeness, timeliness, validity and consistency of both master data (for example, product identifiers, units of measure and warehouse locations) and transactional records (for example, order lines and goods receipts), aligned with the concepts and terminology of the ISO 8000 series on data quality [3]. These requirements are operationalised through a combination of user-interface constraints, server-side validation, reference data dictionaries and database-level constraints, with a clear separation between "preventive" controls (which block incorrect postings) and "detective" controls (which flag anomalies for review).

The persistent layer is implemented on a relational model grounded in contemporary database design principles [4], with a schema that explicitly represents Customers, Suppliers, Products (stock-keeping units), Warehouses, Storage Locations, Orders, Order Lines, Purchase Orders, Goods Receipts, and Stock Movements.

The design prioritises auditability by representing inventory changes as immutable movement records rather than allowing uncontrolled overwrites of stock balances. In practical terms, each operational action that affects availability - receiving, reservation, picking, shipment, return, write-off and adjustment - creates a movement record with a timestamp, actor identity, source document linkage and quantity delta.

The current stock balance is then derived by aggregating movements, optionally materialised in a balance table for performance, while retaining the movement ledger as the source of truth. Transactional integrity is ensured through database transactions that provide atomicity, consistency, isolation and durability (ACID), meaning that a multi-step operation either completes fully or has no effect, and concurrent updates cannot produce partial or contradictory states. To reduce duplication under unstable connectivity or repeated user submissions, critical posting operations are implemented as idempotent commands, allowing the same request to be safely replayed without creating multiple postings.

Concurrency hazards common in stock systems - such as two users attempting to allocate the last available units simultaneously - are addressed through optimistic concurrency controls (e.g., version checks on aggregates) and, where necessary, short-lived locks on specific stock lots or reservation records.

The application is implemented as a web-based client–server solution with a modular architecture that separates presentation, application services, domain logic, and persistence. The order module supports registration, editing under controlled states, and status transitions (created, confirmed, allocated, picked, shipped, closed), with state-dependent validation to prevent inconsistent actions. The inventory module supports goods receipt against purchase orders, cycle counting, adjustments with mandatory reasons, and automated alerts when reorder thresholds are reached. The procurement module provides supplier catalogues, purchase requisitions, and draft purchase orders, informed by reorder logic and historical consumption, while the reporting module provides operational summaries, including stock balances, stock ageing, order backlog, fulfilment lead-time distributions, and discrepancy reports.

Particular attention is given to error prevention in routine human interactions: input forms constrain units of measure, enforce mandatory fields, implement range checks, and provide immediate feedback so that users can correct mistakes at the point of entry rather than through later reconciliations. The system also incorporates structured logging and an audit trail that records who changed what and when, supporting reconciliation and accountability in inventory accounting and internal control environments.

Verification activities are planned and documented in accordance with recognised software test techniques, including equivalence partitioning, boundary value analysis and decision tables, as codified in ISO/IEC/IEEE 29119-4:2021 (International Organisation for Standardisation / International Electrotechnical Commission / Institute of Electrical and Electronics Engineers) [5].

Beyond functional testing, the verification strategy emphasises invariants critical to inventory correctness, such as non-negative available-to-promise quantities, conservation of stock across paired movements (e.g., transfers), and the prohibition of state transitions that would violate business policy (e.g., shipment without allocation). Automated regression suites cover the core workflows (order creation, allocation, picking, receiving, adjustments) and include negative tests for validation, authorisation failures and duplicate submissions.

This test programme is complemented by scenario-driven acceptance tests that reflect realistic small-business operations, such as handling partial deliveries, returns, damaged goods, and backorders, and by data-quality tests that detect orphan records, inconsistent units of measure, and duplicate master data entries.

Information security controls are defined through a risk-based approach that integrates practical web-application guidance with formal management-system requirements and secure development practices. The threat model references the Open Worldwide Application Security Project (OWASP) Top 10:2021 for prevalent categories of web-application risk (including broken access control, injection vulnerabilities and insecure design) [6].

Governance and control selection align with ISO/IEC 27001:2022 (Information Security Management System requirements) [7], ensuring that access control, logging, incident handling, and supplier/dependency management are treated as managed processes rather than ad hoc decisions. Secure engineering practices for the development lifecycle are guided by NIST Special Publication 800-218, the National Institute of Standards and Technology (NIST) Secure Software Development Framework (SSDF), which recommends integrating security activities into requirements, design, implementation, verification, and release [8].

Specifically, the system implements least-privilege access, secure session management, parameterised database queries, server-side input validation, structured error handling that avoids leaking sensitive information, and secure logging with controls to reduce the risk of tampering. Dependency management practices include controlled updates and monitoring for known vulnerabilities in third-party libraries.

Authorisation follows a Role-Based Access Control (RBAC) approach, grounded in the foundational RBAC model literature, to ensure that users can perform only actions consistent with their operational roles (e.g., sales clerk, warehouse operator, supervisor) [10]. This reduces both malicious and accidental misuse, particularly for high-impact actions such as stock adjustments, order cancellations and price changes.

Usability and accessibility are treated as first-class non-functional requirements because the system's primary goal is to reduce manual operations and user error.

The user interface aims to minimise cognitive load through clear task flows, consistent terminology, and specific, actionable error messages presented near the relevant input. Accessibility objectives are mapped to the Web Content Accessibility Guidelines (WCAG) 2.2, a recommendation published by the World Wide Web Consortium (W3C), to support inclusive operation and reduce errors stemming from poor interaction design [9] (e.g., unclear focus indicators, insufficient labelling, or inaccessible form controls).

The engineering process, including iterative refinement, requirements traceability and validation discipline, reflects widely accepted software engineering guidance that links quality outcomes to systematic specification and verification rather than late-stage testing alone [11].

As empirical industrial deployment data are not yet available, an evaluation framework is proposed to quantify effectiveness after pilot introduction. Functional effectiveness is assessed via scenario-based testing that measures completion rates and correctness for typical tasks (order registration, receiving, picking, adjustment, returns). Data quality is assessed using measurable indicators, such as the completeness of mandatory fields, consistency of product identifiers across modules, timeliness of postings, and the frequency of reconciliation discrepancies, with terminology aligned with the data-quality perspective adopted in ISO 8000 [3].

Productivity outcomes are approximated using proxy measures, such as fewer manual ledger updates, shorter time to register deliveries, and reduced rework resulting from incorrect postings. Performance evaluation focuses on response time under concurrent use typical of small businesses, with attention to percentile latency, error rates, and throughput under load conditions that reflect real operational peaks (e.g., end-of-day order processing).

The main limitation of the present study is the absence of long-term operational datasets that would enable statistically robust measurement of economic impact and error reduction across multiple organisations. Integration with external fiscal, point-of-sale or accounting platforms is also out of scope at this stage. Nevertheless, the proposed system design demonstrates a methodologically grounded approach to automating order and inventory accounting in SMEs, with explicit consideration of data quality, transactional integrity, security and accessibility, thereby providing a pragmatic foundation for pilot deployment and subsequent quantitative evaluation.

LIST OF REFERENCES

1. OECD. The Digital Transformation of SMEs : report. — Paris : OECD Publishing, 2021. — 275 p. — DOI: 10.1787/bdb9256a-en. — URL: https://doi.org/10.1787/bdb9256a-en (date of access: 26.01.2026).

2. Muller, P.; Ladher, R.; Booth, J.; Mohamed, S.; Gorgels, S.; Priem, M. et al. Annual report on European SMEs 2021/2022 : SMEs and environmental sustainability : report. — Luxembourg : Publications Office of the European Union, 2022. — ISBN 978-92-9469-291-7. — DOI: 10.2826/451263. — URL: https://op.europa.eu/en/publication-detail/-/publication/0a009ca6-eac2-11ec-a534-01aa75ed71a1/language-en (date of access: 26.01.2026).

3. ISO 8000-1:2022. Data quality — Part 1: Overview. — Geneva : International Organization for Standardization, 2022. — 20 p. — URL: https://www.iso.org/standard/81745.html (date of access: 26.01.2026).

4. Silberschatz, A.; Korth, H. F.; Sudarshan, S. Database System Concepts. — 7th ed. — New York : McGraw Hill, 2020. — ISBN 978-0-07-802215-9.

5. ISO/IEC/IEEE 29119-4:2021. Software and systems engineering — Software testing — Part 4: Test techniques. — Geneva : International Organization for Standardization, 2021. — 135 p. — URL: https://www.iso.org/standard/79430.html (date of access: 26.01.2026).

6. Glas, B. The Release of the OWASP Top 10:2021 [Electronic resource]. — 24 September 2021. — URL: https://www.owasptopten.org/the-release-of-the-owasp-top-10-2021c0a347a0 (date of access: 26.01.2026).

7. ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection — Information security management systems — Requirements. — Geneva : International Organization for Standardization, 2022. — 19 p. — URL: https://www.iso.org/standard/27001 (date of access: 26.01.2026).

8. Souppaya, M.; Scarfone, K.; Dodson, D. Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities : NIST Special Publication 800-218. — Gaithersburg, MD : National Institute of Standards and Technology, 2022. — 36 p. — DOI: 10.6028/NIST.SP.800-218. — URL: https://doi.org/10.6028/NIST.SP.800-218 (date of access: 26.01.2026).

9. Web Content Accessibility Guidelines (WCAG) 2.2 : W3C Recommendation [Electronic resource]. — 05 October 2023. — URL: https://www.w3.org/TR/WCAG22/ (date of access: 26.01.2026).

10. Sandhu, R. S.; Coyne, E. J.; Feinstein, H. L.; Youman, C. E. Role-Based Access Control Models // Computer. — 1996. — Vol. 29, No. 2. — P. 38–47. — DOI: 10.1109/2.485845.

11. Sommerville, I. Software Engineering. — 10th ed. — Boston : Pearson, 2016. — ISBN 978-0-13-394303-0.

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

Лінeвич Роман Юрійович — студент групи 3ПІ-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

Linevych Roman Y — Department of Software Engineering, Vinnytsia National Technical University, Vinnytsia, email: pzmag2022@gmail.com