Development of a corporate information system for automating freight forwarding operations on the .net platform

TOKTORBAEV AIBEK[1], ILIYAZOVA NURZAT[2], BAKIROV MUKHAMMEDALY[3]

[1] Candidate of Physical and Mathematical Sciences, Associate Professor, Osh State University
ORCID: 0009-0007-5255-6683
e-mail: ain7@list.ru
[2] Lecturer, Osh State University
ORCID: 0009-0007-2760-2094
e- mail: iliyazovan96@gmail.com
[3] Master's student, Osh State University
e-mail: bakirovmuhammed10@gmail.com

Abstract: *This paper presents the concept of a corporate information system for automating freight forwarding operations on the .NET platform. The system integrates order registration, cargo tracking, document management, customer communication, and reporting. Human factors, staff responsibilities, and data protection are considered. A hypothetical evaluation scenario, a table, and illustrative charts demonstrate how system performance could be assessed.*

Keywords*: corporate information system, freight forwarding, .NET, database, logistics, business process automation, human factors.*

Introduction

Freight forwarding companies coordinate interactions among customers, carriers, warehouses, drivers and regulatory authorities. For each order, the route, deadlines, cost, documents and current cargo status must be agreed upon. When this information is kept in separate spreadsheets, emails and messengers, the risk of data duplication, delayed status updates and document errors increases. A corporate information system (CIS) makes it possible to integrate these processes and create a single source of up-to-date information [1; 2].

The aim of this work is to propose a CIS model for freight forwarding operations on the .NET platform and to define indicators for assessing its usefulness. The objectives include process analysis, requirements definition, architecture and database design, consideration of human factors, and development of a performance evaluation plan.

The practical significance lies in reducing repeated data entry, increasing order transparency and simplifying management reporting. At the same time, automation does not eliminate human oversight: employees confirm critical operations, review exceptions and make decisions in non-standard situations.

Domain Analysis

A typical order cycle includes receiving a request, verifying customer and cargo data, calculating the cost, selecting a carrier, planning the route, preparing documents, tracking the transportation stages, confirming delivery and closing the order. At each stage, information is generated that is available to authorized employees and, to an established extent, to the

customer.

The main problems of manual management include repeated data entry, lack of a complete status history, difficult document retrieval, inconsistency between the manager, dispatcher and accounting department, reporting delays and the risk of unauthorized access. Consequently, the system must support centralized storage, action logging, access rights separation and notifications.

System Requirements

Functional requirements: registration of customers and counterparties; creation and modification of orders; storage of cargo characteristics; assignment of a carrier and a manager; management of delivery stages; document upload; accounting of expenses and payments; notifications of important events; generation of reports on deadlines, costs and statuses.

Non-functional requirements include role-based access control, backup, input data validation, logging of key actions, acceptable response time and extensibility. For personal and commercially sensitive data, retention periods, access procedures and incident response should be defined [3; 4].

Solution Architecture on the .NET Platform

The proposed architecture includes four logical layers: the user interface, application logic, data access, and the database with infrastructure services. For a desktop workstation, C# with WPF or Windows Forms can be used; for multi-user web access, ASP.NET Core. The choice depends on the size of the company, operating conditions and the need for remote work.

The application layer implements order processing rules, validation of status transitions, assignment of responsible staff, calculation of indicators and report generation. Data access can be organized using Entity Framework Core or parameterized SQL queries, with Microsoft SQL Server as the DBMS. Separating the layers facilitates testing and maintenance.

The main database entities are Client, Order, Cargo, Route, Carrier, Employee, OrderStatusHistory, Document, Expense, Payment and Notification. Each order has a unique identifier; the status history records the time of change, the previous and new states, and the user who performed the action. Relationships are defined by foreign keys, and critical fields are protected by integrity constraints.

Logical Data Model

Entity	Main fields	Purpose
Client	ClientID, Name, Contact	Customer information
Order	OrderID, ClientID, Date, Status	Order record
Cargo	CargoID, OrderID, Weight	Cargo characteristics
Carrier	CarrierID, Name, Contact	Carrier
Document	DocumentID, OrderID, Type, URI	Order documents
StatusHistory	HistoryID, OrderID, Status, Time, UserID	Status history
Expense/Payment	ID, OrderID, Amount, Date	Expenses and payments

Table 1. Example of the main database entities.

Order Life Cycle

It is recommended to define the permitted order states: “New”, “Verified”, “Scheduled”, “In Transit”, “Delivered”, “Closed” and “Cancelled”. Transitions are validated by the application logic. For example, an order cannot be closed without delivery confirmation and the mandatory set of documents, if required by the regulations.

Consideration of Human Factors

The effectiveness of a CIS depends not only on the software but also on how well employees understand the process and are willing to work in a shared system. Resistance to implementation is possible if the interface is complex, training is insufficient or new operations are perceived as an additional burden. Therefore, human factors should be taken into account at the design, testing and launch stages.

It is proposed to separate roles: the

administrator manages user accounts; the manager registers requests and communicates with customers; the dispatcher plans shipments and updates stages; the accountant records expenses and payments; the executive reviews summary reports. The principle of least privilege reduces the risk of accidental or intentional data modification.

User support requires instructions, training on typical scenarios, a test environment and a feedback channel. It is advisable to measure the share of correctly completed orders, the number of help requests, the number of corrections and user satisfaction. Changing the cost or cancelling an order may require confirmation and a mandatory entry in the audit log.

6. Example of Performance Evaluation

For illustration, consider a hypothetical scenario of order processing before and after automation. The values of time, labor effort and on-time delivery below are approximate and are not data from a real company. For a practical evaluation, baseline indicators must be recorded, a pilot implementation carried out, and measurements repeated with a comparable volume of orders.

Process	Before, min	After, min	Change, %
Order registration	18	7	−61.1
Shipment planning	35	18	−48.6
Status update	22	9	−59.1
Report preparation	28	12	−57.1

Table 2. Hypothetical processing times of operations.

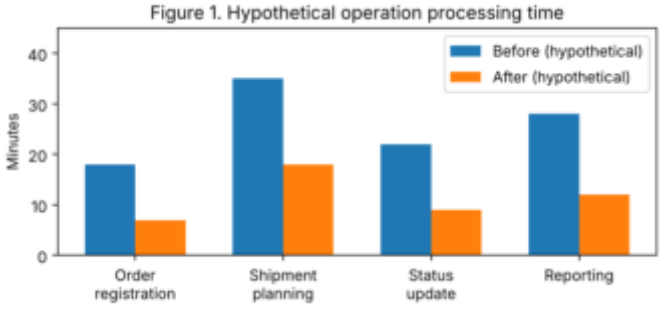


Figure 1. Hypothetical comparison of operation times before and after automation.

In this example, the potential reduction in time is explained by less repeated data entry, the use of shared reference data and automatic report generation. In a real project, the effect depends on seasonality, cargo complexity, the quality of source data and employee experience.

Labor Effort Indicators

To analyze human factors, the distribution of working time among manual data entry, route planning, document checking and customer communication can be compared. Automation may reduce the share of repetitive operations but increase the need for exception handling and data quality control.

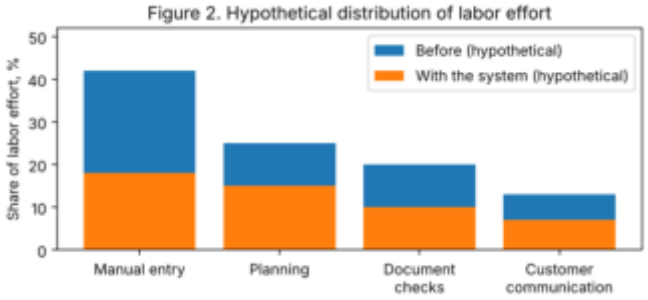


Figure 2. Hypothetical distribution of labor effort by task.

The chart illustrates a way of presenting the results but does not prove an actual reduction in workload. Verification requires measurements or event logs before and after implementation using the same rules for classifying operations.

Key Performance Indicators

For monitoring, the following KPIs are proposed: the share of orders delivered on time; average request registration time; document processing time; the share of orders with a complete set of documents; the number of errors per 100 orders; the number of overdue status updates; employee and customer satisfaction. For each indicator, the formula, data source, calculation period and responsible employee must be defined.

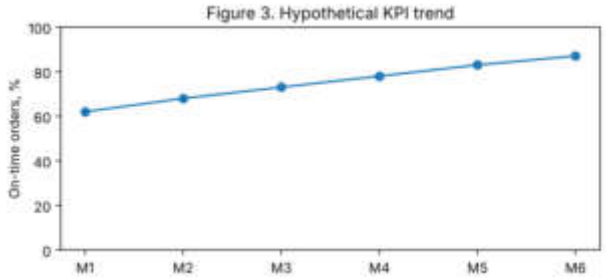


Figure 3. Hypothetical trend in the share of orders completed on time.

The line reflects only an illustrative scenario.

On-time delivery also depends on road conditions, vehicle availability, weather, the actions of counterparties and the accuracy of planning. Therefore, not all changes can be attributed solely to the implementation of the software.

Information Security and Reliability

The CIS stores commercial information, customer contacts, transportation data and financial records. Protection measures include authentication, role-based authorization, secure connections, secure storage of secrets, backup, component updates and security event logging. Passwords must not be stored in plain text; accounts of dismissed employees should be blocked in a timely manner.

It is necessary to define the backup frequency, the storage location of copies and the recovery time objective, and to test recovery regularly. Protection against SQL injection, checking of uploaded documents and restrictions on file types and sizes are recommended. When processing personal data, applicable legislation and internal company rules must be taken into account [3; 4].

Implementation and Testing Plan

Implementation should be carried out in stages: process survey; requirements gathering; architecture and database design; development of the minimum required functionality; unit and integration testing; verification of access rights and security; pilot operation; training; KPI analysis and gradual expansion. Before launch, acceptance criteria and a rollback plan to the previous process in case of critical failure are defined.

Testing should cover correct and erroneous scenarios, concurrent user activity, failure recovery, access control and data integrity. Particular attention is paid to order cancellation, route changes, cargo delays, incomplete document sets and discrepancies in financial data. Employee feedback is used to improve the interface and instructions.

Conclusion

This paper has considered the concept of a corporate information system for automating freight forwarding operations on the .NET platform. A multi-layer architecture has been proposed, and the main modules, database entities, order life cycle and role-based user permissions have been defined. Human factors, staff training, action auditing and data protection have been considered separately.

The tables and charts show how an assessment of the potential effect of automation can be organized; however, all the values presented are hypothetical. To confirm effectiveness, a pilot implementation must be carried out, the baseline and final KPIs measured, and the experimental conditions described. Future work may include integration with GPS/telematics, electronic document management, notifications and external carrier services.

References

1. Laudon K. C., Laudon J. P. Management Information Systems: Managing the Digital Firm. 16th ed. Pearson, 2020.
2. Sommerville I. Software Engineering. 10th ed. Pearson, 2015.
3. ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection — Information security management systems — Requirements. Geneva: ISO, 2022.
4. NIST. The NIST Cybersecurity Framework (CSF) 2.0. NIST CSWP 29. Gaithersburg, MD: National Institute of Standards and Technology, 2024.
5. Microsoft. .NET documentation. Microsoft Learn. https://learn.microsoft.com/dotnet/:
6. Microsoft. SQL Server documentation. Microsoft Learn. https://learn.microsoft.com/sql/.