http://doi.org/10.35784/iapgos.7639 *received: 06.02.2025 | revised: 22.09.2025 | accepted: 18.11.2025 | available online: 20.12.2025*

ANALYSIS OF THE INTERACTION OF COMPONENTS OF A MODULAR PARCEL STORAGE SYSTEM USING UML DIAGRAMS

Lyudmila Samchuk[1], Yuliia Povstiana[2], Anastasia Hryshchuk[2]
[1]Lutsk National Technical University, Faculty of Transport and Mechanical Engineering, Department of Applied Mechanics and Mechatronics Lutsk, Ukraine,
[2]Lutsk National Technical University, Faculty of Computer and Information Technologies, Department of Software Engineering, Lutsk, Ukraine

Abstract. *The article studies the interaction of components of a modular parcel storage system using UML diagrams. The key aspects of modeling the structural and dynamic interaction of components, such as physical elements (cells, scanners, conveyors) and software modules (control system, database), to ensure flexibility, scalability and automation of logistics processes are considered. The author presents the use of class diagrams, sequences, precedents, and activities to detect logical errors, predict potential failures such as overload or out-of-sync, and optimize system performance. Particular attention is paid to the roles of actors (client, manager, designer, logistician, etc.) and their interaction with the system, which allows for a clear definition of functional requirements and use cases. The modeling results demonstrate the possibility of improving the efficiency of processing, tracking, and delivery of parcels, as well as providing convenient access for users. The data obtained can be used to improve modern solutions in the field of automated parcel management systems, contributing to the creation of reliable and adaptive logistics platforms that can meet the growing demands of the market.*

Keywords: modular system, parcel storage, UML diagrams, component interaction, system architecture, class diagram

ANALIZA INTERAKCJI ELEMENTÓW MODUŁOWEGO SYSTEMU PRZECHOWYWANIA PACZEK Z WYKORZYSTANIEM DIAGRAMÓW UML

Streszczenie. *Artykuł przedstawia interakcję elementów modułowego systemu przechowywania paczek przy użyciu diagramów UML. Kluczowe aspekty modelowania strukturalnej i dynamicznej interakcji komponentów, takich jak elementy fizyczne (komórki, skanery, przenośniki) i moduły oprogramowania (system sterowania, baza danych), w celu zapewnienia elastyczności, skalowalności i automatyzacji procesów logistycznych. Autor przedstawia wykorzystanie diagramów klas, sekwencji, precedensów i działań do wykrywania błędów logicznych, przewidywania potencjalnych awarii, takich jak przeciążenie lub brak synchronizacji, oraz optymalizacji wydajności systemu. Szczególną uwagę zwraca się na role aktorów (klient, menedżer, projektant, logistyk itp.) i ich interakcję z systemem, co pozwala na jasne zdefiniowanie wymagań funkcjonalnych i przypadków użycia. Wyniki modelowania pokazują możliwość poprawy wydajności przetwarzania, śledzenia i dostarczania paczek, a także zapewnienia wygodnego dostępu dla użytkowników. Uzyskane dane mogą zostać wykorzystane do ulepszenia nowoczesnych rozwiązań w dziedzinie zautomatyzowanych systemów zarządzania przesyłkami, przyczyniając się do stworzenia niezawodnych i adaptacyjnych platform logistycznych, które mogą sprostać rosnącym wymaganiom rynku.*

Słowa kluczowe: system modułowy, przechowywanie paczek, diagramy UML, interakcja komponentów, architektura systemu, diagram klas

Introduction

Modern logistics systems, including parcel storage systems, face the challenge of providing a high level of automation and efficient interaction between components for fast processing, accurate tracking, and easy access to parcels. Modular systems, which include physical elements such as cells, scanners, or conveyors, and software modules such as a management system or database, require a clear design to guarantee flexibility, scalability, and reliability. However, the lack of clear mechanisms for interacting with these components can lead to incorrect implementation, inefficient use of resources, difficulty expanding the system, cell overload, synchronization failures, or delays in data processing, making it difficult to adapt to changing conditions and growing workloads.

UML diagrams solve these problems by providing a tool for creating a coherent and efficient solution. They visualize the structure of a system by showing the relationships between physical and software components, and allow you to describe the roles of each – for example, parcel identification by a scanner or coordination through a software controller – while avoiding duplication of functions. Modeling dynamic behavior, such as the process of receiving or issuing a parcel, helps to identify logical errors or delays before implementation, and analyzing states and data flows predicts potential failures such as overload or out-of-sync, allowing you to eliminate them at the design stage. UML provides flexibility and scalability, making it easy to integrate new modules without compromising integrity, and optimizes the placement of elements, such as servers, to increase efficiency. In this way, UML guarantees automation, accurate real-time tracking, convenient user access, and reliability through redundant mechanisms, creating a system that can adapt to modern logistics challenges.

The objective to analyze and model the processes of interaction between the components of a modular parcel storage system using UML diagrams in order to identify possible optimizations and improve its functionality.

The introduction outlines the need for automation and efficiency in modular parcel storage systems, highlighting challenges like resource inefficiency and scalability issues. UML diagrams are presented as a solution to visualize structure, model behavior, and ensure flexibility, with the goal of optimizing system interactions and functionality.

1. Literature review

The authors of the article [10] "Parcel storage using Internet of Things (IoT) technologies", in particular Jing Zhi Ooi and others, in their work, offer an innovative solution to automate the last mile of parcel delivery. They focus on the use of IoT to create a modular system of smart lockers that allows for contactless placement and retrieval of parcels, increasing the efficiency of logistics processes.

The article addresses the problem of rising costs for logistics companies due to the increase in delivery endpoints. The proposed system of modular lockers allows consolidating parcels for easy retrieval. An innovative method for measuring the size of objects is presented.

The analysis of research shows the active development of the topic of smart lockers, in particular in the aspects of security, scalability, and IoT integration. Compared to the works on consumer perception, the article focuses on the technical implementation and implementation of IoT in logistics.

The study is a significant contribution to the field of last-mile delivery, but requires further improvements in terms of cost optimization, security, and adaptation to different markets.

Summary. The paper [10] proposes an IoT-oriented solution for automating delivery through smart postal machines, focusing on technical feasibility and efficiency. It needs to be improved in terms of security, cost, and adaptation to markets.

A research article [14], authored by representatives of Parcel Pending Inc, describes a system of smart lockers for delivering and receiving parcels. The proposed solution is based on the use of electronic lockers with controlled locks connected to an internal computer via input/output devices. The authors emphasize the convenience, security, and modularity of the system, which allows it to be adapted to different needs.

The study focuses on solving the problems of parcel delivery in multi-user environments, offering an alternative to traditional methods that depend on cooperation with online retailers or courier services. Compared to studies such as "Smart Modular Parcel Locker System using IoT", which focuses on IoT and dimensional measurement, this paper is more focused on practical implementation and access control. The authors point out the limitations of previous systems (dependence on partnerships or low adaptability) and propose a solution that needs further improvement towards integration with wider logistics networks.

The paper [14] describes a modular system of smart post offices with an emphasis on practical implementation and security, but needs to improve integration with logistics networks.

The article [4] "Parcel management system" by Kamesh Behera discusses the creation of a web-based system for automating logistics operations. The main goal of the study is to simplify parcel delivery processes, reduce paperwork, and increase accuracy and efficiency in parcel management. The system allows both the sender and the recipient to track the status of the parcel, minimizing the risk of losing the parcels thanks to an automated notification system that informs about the delivery status.

The article addresses the problem of unreliability and low efficiency of manual parcel management systems, which often leads to lost parcels, delivery delays, and customer dissatisfaction. Among the key challenges outlined in the study are the need to eliminate human error, ensure accurate accounting and transparency, and reduce costs through process automation.

The system's functionality includes a user-friendly graphical interface, integration with a database for storing parcel information, and the ability to send automatic notifications to users about the status of a shipment. Senders can create delivery requests, and recipients receive updates through their dashboards. All parcel movement data is recorded in the database to ensure transparency and reliability.

The study emphasizes the importance of using modern web technologies to automate logistics processes, which can significantly reduce company costs and increase customer confidence in services. Thus, the study not only points out the current problems of the industry, but also offers an effective solution to overcome them.

Paper [4] presents a web-based system for automating parcel management that increases efficiency and transparency, but emphasizes the need to reduce human error and costs.

The authors of the article [16], Yuxiang Zeng, Yongxin Tong, and Lei Chen, investigate the problem of efficient organization of last-mile delivery. The main problem addressed in the article is the complexity of optimizing delivery routes in real-world conditions, given limited resources, variability in customer requests, and the need to minimize both total waiting time and maximum courier time. Their approach is based on the use of an innovative method, the Hierarchically Separated Tree, which can significantly improve the computational efficiency of the algorithms.

The analysis shows that the proposed algorithms are highly accurate and efficient in route planning. The authors achieve significant improvements over existing methods, especially in scenarios with a large number of queries. In addition, their approach provides theoretical performance guarantees, making it attractive for implementation in real-world delivery systems.

However, the method requires some improvements. For example, the article notes that more research is needed to adapt the algorithm to real-time dynamic changes, such as traffic jams, changes in customer requests, or changes in weather conditions. In addition, the integration of the system with existing delivery platforms may require additional modifications to ensure its practical implementation.

Paper [16] proposes an effective algorithm for optimizing delivery routes, but needs to be adapted to dynamic conditions and broader integration.

The study in [7], authored by Stanisław Iwan, Kinga Kijewska and Justyna Lemke, investigates the effectiveness of automated mailboxes introduced by InPost in Poland. The paper focuses on how these innovative deliveries are made.

The analysis shows that the introduction of mailboxes significantly increases the efficiency of delivery. For example, InPost couriers were able to deliver up to 600 parcels per day with a total route of 70 km, which is significantly higher than traditional delivery methods, where only 60 parcels per 150 km are served. This organization of work significantly reduces fuel consumption and carbon emissions.

The system needs some improvements, such as optimizing the location of mailboxes in urban areas to ensure maximum accessibility for customers. In addition, closer cooperation with local authorities and encouragement of online retailers to use this infrastructure is needed. The authors also emphasize the importance of involving residents in supporting such solutions.

The study [7] demonstrates the high efficiency of InPost post offices, but points to the need to optimize their location and cooperation with partners.

The authors of the article [6], Jairo A. Cortes M., Luis Carlos Gutierrez, Jaime Alberto Paez Paez, Fredys A. Simanca H., and Fabian Blanco Garrido, investigate a system for storing software quality metrics associated with UML diagrams. The main problem addressed in the article is to ensure early detection of errors in the software concept through the use of quality metrics that integrate with UML diagrams. The authors emphasize the importance of creating tools that will help software designers and developers assess the compliance of a diagram with quality standards.

The analysis shows that the system allows you to effectively store information about quality metrics, providing easy access to it through a repository developed on the basis of XMI (XML Interchange Format). This provides a convenient exchange of metadata between different tools for modeling UML diagrams and allows developers to quickly identify deficiencies in development [13].

Improvements that the system needs include adapting it to dynamic changes in software requirements, as well as improving integration with Object Constraint Language (OCL) standards. In addition, the authors note the need to improve the usability of the user interface to expand the use of the system among developers.

The paper [6] proposes a system for assessing the quality of UML diagrams, but needs to be improved to adapt to changes and usability.

The author of the article [2], Beauden John, explores the relationship between the implementation of UML (Unified Modeling Language) and software complexity. The main issue addressed in the article is how the use of UML diagrams affects the reduction of project complexity, improvement of code modularity, and reduction of errors during development.

The analysis shows that the implementation of UML improves the understanding of complex systems and facilitates communication between stakeholders. For example, the use of UML diagrams such as class diagrams and sequence diagrams helps to clearly visualize the system architecture, which helps to reduce the cognitive load on developers. According to the study, it also reduces the risk of design errors, improves the quality of decision-making, and provides more sustainable architectural solutions.

However, the author notes that over-reliance on UML can cause additional difficulties. For example, creating overly detailed diagrams can lead to unnecessary time and difficulties in maintaining them, especially in a dynamic project. The article emphasizes the importance of a balanced approach to using UML, which involves choosing only those artifacts that really contribute to simplifying development and reducing complexity.

Article [2] emphasizes the effectiveness of UML in simplifying development, but emphasizes the need for a balanced approach.

Summarizing the approaches of different authors shows that UML is a powerful tool for managing software complexity, but its effectiveness depends on the correct application. The use of UML should be strategically directed and adapted to the specifics of the project.

In Ukraine, research in the field of modular parcel storage systems and their integration with the Internet of Things (IoT) is not yet widely represented, but there are developments in the field of automated logistics systems. Some domestic scholars consider the development of logistics technologies in the context of the introduction of contactless delivery, including the use of electronic mailboxes.

In particular, papers published in the scientific journals of NTU KhPI and Kyiv National University of Construction and Architecture focus on the creation of intelligent systems for automated parcel processing. The research analyzes ways to optimize the operation of post offices, use RFID tags and QR code technologies to improve security and user convenience.

In addition, domestic logistics companies, such as Nova Poshta and Ukrposhta, are actively implementing automated delivery systems through post offices. Their innovative solutions are aimed at reducing the burden on courier services and increasing the efficiency of delivery in urban areas [12].

At the international level, research in this area is far ahead of domestic research. In particular, the article "Smart Modular Parcel Locker System using IoT" (Jing Zhi Ooi et al.) presents an IoT-oriented modular locker system for automating the last mile of delivery. The main advantage of the proposed solution is the integration of sensor technologies to track the size and condition of parcels in real time.

Another study presented by Parcel Pending Inc [11] analyzes the practical implementation of smart lockers using controlled electronic locks. The automated system allows lockers to be adapted to the needs of a multi-user environment, which is an important aspect when developing modular solutions for business and residential complexes.

The EU countries are actively implementing technologies that can improve logistics processes using artificial intelligence and IoT. For example, research published in the IEEE focuses on integrating machine learning to predict the optimal use of lockers and improve delivery security through blockchain technologies.

It's also worth mentioning the experience of China, where modular parcel storage systems are already being actively used in smart cities. Companies such as Cainiao Network (a subsidiary of Alibaba) are implementing high-tech solutions to automate and scale the delivery process [5].

Thus, a comparison of domestic and foreign studies shows that modular parcel storage systems in Ukraine are still at the stage of active implementation, mainly through private initiatives of logistics companies. While foreign research already covers IoT, artificial intelligence, and blockchain, in Ukraine the focus is on basic automation and integration with existing supply chains.

To improve the situation in Ukraine, it is necessary to intensify research in the field of UML modeling of the interaction of components of such systems, which will increase the efficiency of designing and implementing smart post offices in the domestic logistics infrastructure. Logistics automation is developing in Ukraine, but lags behind international IoT and AI solutions. UML modeling research is needed to improve efficiency.

2. Researches methodology

The methodology of the study is based on the application of UML diagrams – including use case, class, sequence, and activity diagrams – to model the structure, dynamics, and logic of interaction between components of the modular parcel storage system. This approach enables the identification of logical inconsistencies at the design stage, optimization of design, manufacturing, and delivery processes, and ensures consistency between the physical and software modules of the system.

The effectiveness of the proposed approach is evaluated using key performance indicators (KPI), such as the number of modeled use cases (at least ten), the number of architectural layers (not fewer than three), the scalability index of the system (more than 80 % of modules can be integrated without modification), and the number of detected modeling errors (no more than three). The obtained results confirm the increased flexibility, reliability, and performance of the system, ensuring its efficient adaptation to the growing demands of the logistics sector.

3. Results

In modern logistics systems, automated modular parcel storage plays an important role in ensuring efficient management of the storage and delivery of parcels. Analyzing the interaction of their components is key to optimizing system performance. The use of UML diagrams allows you to visualize the structure and dynamics of the interaction of elements, which helps to improve the software architecture and its efficiency [3].

This paper proposes the development of four UML diagrams: precedents, sequences, activities, and classes [9].

- the precedence diagram will help define user roles and key scenarios of interaction with the system;
- a sequence diagram will allow you to analyze the order of message exchange between objects;
- the activity diagram will show the flow of processes in the system and their logic;
- a class diagram will define the structure of the system, its main classes, attributes, and relationships.

The use of these UML diagrams will help identify potential problems, optimize logistics processes, and ensure the flexibility and scalability of the system.

The first diagram is the precedence diagram (Fig. 1), which shows the interaction between actors and the system, as well as all key use cases [15]. At the same time, the focus is not only on the UML precedence diagram, since all of the listed diagrams – precedents, sequences, activities, and classes – will be described for a comprehensive analysis of the system. This approach allows you to study in detail the full cycle of interaction – from ordering a post office machine by a customer to its delivery and subsequent support. The use of different types of UML diagrams helps to identify potential problems at the design stage, optimize logistics processes, and ensure the flexibility and scalability of the system. The analysis is based on the roles of actors, their interactions with the system, and the structure and dynamics of the system, which is key to creating a reliable and effective solution.

The following UML precedent diagram depicts the interaction between actors and the system in the context of developing, ordering, manufacturing, and delivering postal machines. The actors include the customer, sales manager, designer, engineer, programmer, logistician, and warehouse, each of whom performs certain functions. The main precedents cover the processes of ordering, developing, manufacturing, delivering, and supporting postal machines.

Actors and their interaction form the basis of the system's functioning. The customer participates in the processes related to ordering, payment, design selection, feedback, and authorization, having the opportunity to order a post office both online and offline, with further agreement on the details. The sales manager handles order processing, invoicing, negotiating delivery terms, and keeping in touch with the client, following the transaction from start to finish, including delivery times. The designer focuses on developing the design, approving it with the client, and making the necessary changes, where the relationship between these actions is supported by dependencies. The engineer is responsible for the technical aspects, such as developing specifications, adapting the design, and supporting production, with the basic design as the foundation for the subsequent stages. The programmer is responsible for creating and updating software, as well as integrating with payment systems. The logistician coordinates the planning, tracking, and coordination of delivery times, while the warehouse manages inventory, receiving, storing, and preparing products for shipment.

The relationships between use cases reflect their logical dependence. Dependencies of type <<include>> indicate mandatory actions, for example, "Order a post office" requires "Pay for the order" and "Registration and authorization" to complete. The relationships <<extend>> indicate additional scenarios, for example, "Design selection", which can be involved after the order is completed at the customer's request. The generalization is applied to "Online order" and "Offline order", which are variations of the basic use case "Order a post office".

The main use cases cover the entire interaction cycle: from the initiation of the order by the customer to its delivery and subsequent support. The process includes sequential actions – from order processing by the manager, design development by the designer, creation of the structure by the engineer, programming by the programmer to organizing delivery by the logistician and storage in the warehouse with subsequent shipment of finished products.

The use case diagram helped to identify the key users of the system and their interaction with the system. However, for a more detailed representation of the internal structure of the system and the relationships between its components, it is necessary to turn to a class diagram (Fig. 2). It will allow you to clearly define the data structure, attributes, operations, and relationships between objects, which is critically important for the further implementation of the system [8].

The second diagram, presented as a UML class diagram, illustrates the static structure of a modular parcel storage system, defining classes, their attributes, operations, and relationships between them. This diagram displays key components of the system, such as customers, sales managers, designers, engineers, programmers, logisticians, warehouse, and orders, and their interactions that form a coherent architecture. Client displays customer data, including ID, authorization status, first name, last name, contact information, address, and payment card information, with operations that allow you to retrieve this data and search by ID. SalesManager contains information about the sales manager, including ID, first name, last name, phone number, and number of active orders, providing methods for accessing the data, creating orders, and preparing quotes. Designer represents a designer by name, surname, contact number, ID, specialization, and number of projects, with operations for developing, updating, and approving designs. Engineer represents an engineer by name, surname, ID, specialization, and number of tasks, providing for the creation of technical specifications, testing, and modification of designs. Programmer represents a programmer by name, surname, ID, skills, and number of projects, with operations for developing software, integrating payment systems, and updating it. Logistician represents a logistician by ID, surname, name, phone number, and number of deliveries, providing capabilities for planning and tracking deliveries. Warehouse represents a warehouse by ID, location, capacity, current inventory, and storage conditions, with operations for receiving and preparing mailers for shipment. Order defines an order via ID, customer ID, date, status, payment details, and delivery date, with operations to access information, update status, and confirm payments.

The relationships between classes provide integration and coordination in the system. Order is related to Client through the clientID attribute, which reflects the dependence of the order on the client. SalesManager interacts with Order by creating new orders, and Designer and Engineer participate in the development of design and construction, using the corresponding operations to implement projects. Programmer supports the software component by developing and updating software, while Logistician and Warehouse coordinate logistics processes, including planning, tracking and preparation for shipment.

The functional structure of the class diagram reflects the hierarchy and distribution of responsibilities between participants, where each class has clearly defined attributes and methods that ensure the continuity of processes – from order processing to its delivery. Relationships between classes emphasize the unity of physical and software elements, contributing to effective interaction and the formation of a reliable modular parcel storage system.

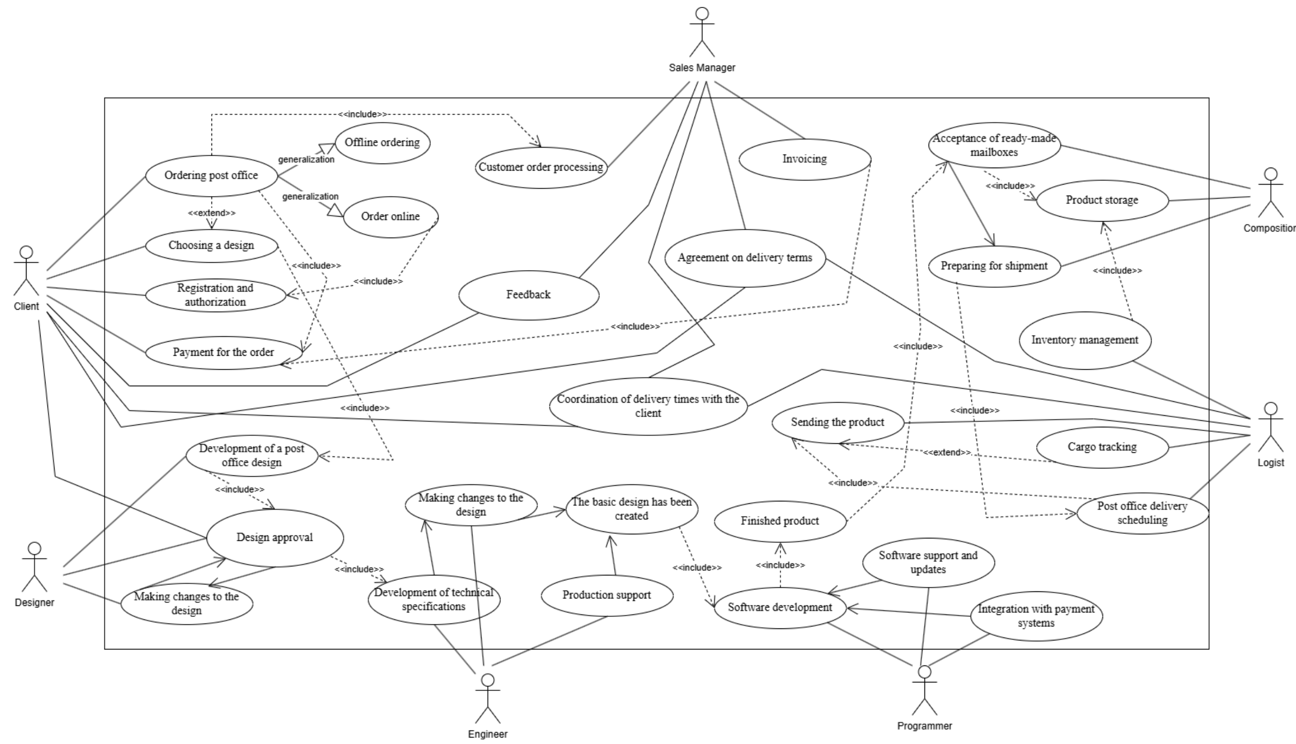

Fig. 1. UML diagram of precedents

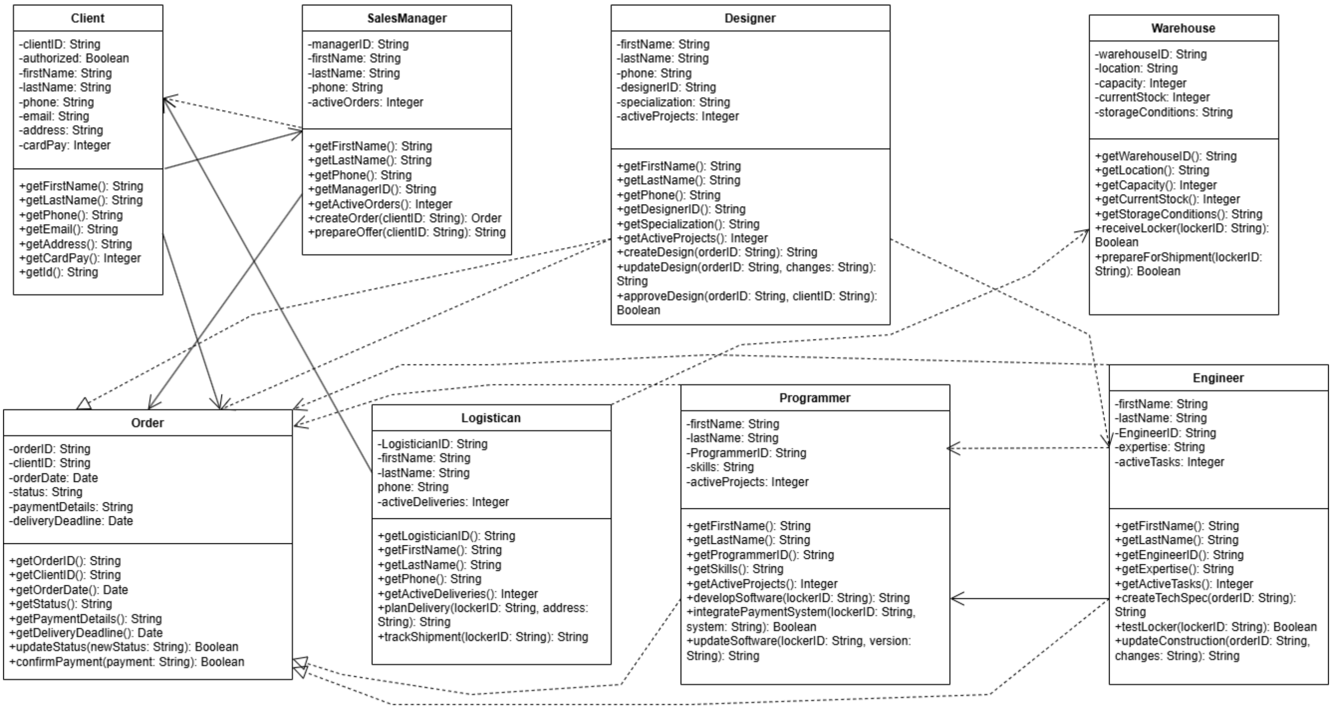

Fig. 2. UML class diagram

The class diagram provides an understanding of the structure of the system, however, for a detailed analysis of its dynamic behavior, it is necessary to consider the sequence diagram (Fig. 3). It will help visualize the order of interaction between objects and actors during basic business processes, such as ordering, developing, and delivering mail machines. This will allow identifying possible communication problems between system components and optimizing their interaction [17].

The diagram shows the chronological order of actions and messages exchanged between actors in the process of order fulfillment, development and delivery of POS machines. It demonstrates the dynamic interaction between key participants – the customer, the online platform, the sales manager, the designer, the engineer, the programmer, the logistician and the warehouse – from the initial stage to the completion.

The process begins with the customer's registration and authorization via the online platform, after which the customer selects the POS machine model and agrees the design with the designer, receiving his approval. Next, the customer fills in the order data, which is transferred to the sales manager via the platform, where an invoice is generated for payment. After payment, the manager coordinates the technical development, involving an engineer to create specifications and a programmer to develop the software. The finished product is transferred to the warehouse, from where the logistician plans the delivery, coordinating the terms with the customer and providing tracking. The customer receives information about the order status and delivery confirmation, completing the cycle.

The interaction between objects is built as a clearly structured chain, where each message reflects a separate stage of the process, from the initial interaction to the logistical implementation. This approach allows you to visualize the sequence of actions, identify possible points of delays or failures, as well as optimize coordination between participants, which helps to increase the efficiency and reliability of the entire system. The fourth diagram is an activity diagram (Fig. 4), which reflects the logic of work processes and the sequence of operations in the system, providing a clear understanding of the stages of task execution from ordering a post office to its delivery and subsequent support. It consists of three diagrams: an activity diagram for the complex system process; an activity diagram of the event flow of the post office order processing process from the customer's point of view; an activity diagram of the event flow of the process of monitoring existing orders by the manager.

The sequence diagram clearly demonstrates communication between system participants, but for a more detailed analysis of internal business processes, it is advisable to consider an activity diagram (Fig. 4), in which there are seven subsystems, each of which performs its own role: client, sales manager, designer, engineer, programmer, logistician, warehouse. It helps not only to trace the logic of task execution, but also to analyze options for their optimization, minimizing unnecessary actions and improving the overall productivity of processes [1].

This diagram reflects the logic and sequence of operations covering the entire cycle of creation, processing and delivery of post-office machines, providing a holistic understanding of the interaction between participants. The process begins with the initiation of an order by the client, who chooses the method of submission – online via the website or offline via the manager. The manager accepts the order, advises the client on the product's capabilities, creates a commercial offer and issues an invoice, after which the client makes a payment. Next, the manager passes the design requirements to the designer, who develops a sketch, agrees with the client and makes changes if necessary, transferring the approved version to the engineer. The engineer creates technical specifications and a basic design, collaborating with the programmer, who develops software and integrates it with payment systems. After testing the operability, the engineer transfers the finished product to the warehouse, where it is registered. The logistician plans the delivery, agrees on the terms with the client through the manager, tracks the cargo and delivers the post-office machine, completing the process. The client has the opportunity to leave feedback, which can initiate design adjustments.

To better understand the process of customer interaction with the system, it is advisable to consider the activity flow diagram of the post-machine ordering process from the customer's perspective (Fig. 5). It shows the key stages – from logging into the system to viewing the catalog of available post-machines, selecting a model and specifications, placing an order, making payment, and receiving the finished product.

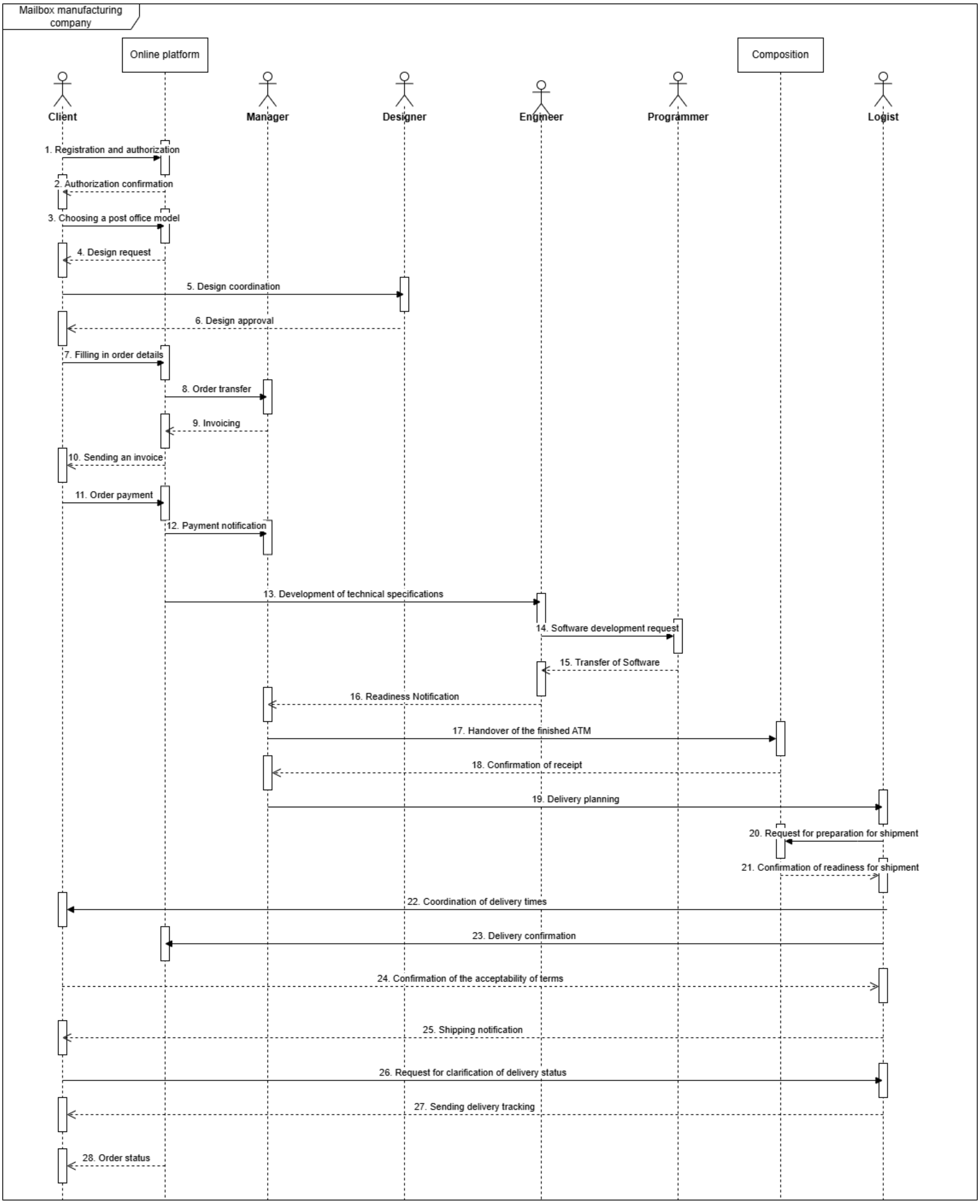

Fig. 3. UML sequence diagram

After entering the necessary information, including choosing a design, payment method, and delivery, the customer confirms the order online through the system. Ultimately, the customer pays for the order and receives their post-machine.

The diagram illustrates the process from the customer's perspective, starting with their registration or authorization on the company's website, where they gain access to a catalog of available ATM models with specifications and prices. The customer then selects a model, defines design and functionality requirements, fills out an order form with delivery details, contacts and payment method, choosing between a card or bank transfer, and the delivery method. After confirming the order, the system automatically transfers it for processing, generates an invoice and sends it to the customer, who makes the payment via an integrated payment system. The order goes into production, where the system notifies the responsible specialists about the start of production, tracking the stages of creating the design, construction and software. After production is complete, the system transfers the data to the logistician to prepare for delivery, which ends with the customer receiving the ATM. At the final stage, the system requests confirmation of receipt and offers to leave a review, which is added to the company's website.

Similarly, a flow chart of the process of monitoring orders by the manager is constructed (Fig. 6). It covers the entire cycle of working with orders: acceptance, review, status update, tracking of execution stages and confirmation of receipt by the client. The manager has access to the corporate system, where he reviews and processes orders from post offices from the moment they are placed (online or offline) until the moment of delivery. This ensures timely processing of orders, up-to-date status updates and prompt information to customers.

The activity diagram illustrates the process of the manager monitoring orders for post-office machines, starting from the moment they are placed by the customer until the finished product is received. The manager authenticates in the company system, then reviews the list of active orders that could have been placed online or offline. If the order is new, he confirms its acceptance and checks the payment status. If the payment is confirmed, the manager updates the status in the system, indicating the stages of processing, design, production or other intermediate processes. Throughout the process, the manager maintains contact with the customer, clarifying details, agreeing on changes or handling complaints. In parallel, he coordinates the work of the team, transferring data to designers, engineers and programmers, and monitors the implementation of each stage. To organize delivery, the manager checks the availability of a finished post-office machine in the warehouse, after which he transfers the order to the logistician and agrees on the details with the customer. After the customer receives the post-office machine, the manager confirms the completion of the order and collects feedback for further analysis and improvement of the service.

The development and analysis of UML diagrams – precedents, classes, sequences, and activities – allowed us to comprehensively study the modular parcel storage system, covering its structure, interaction dynamics, and process logic. The precedent diagram defined the roles of actors and use cases, the class diagram outlined the static architecture, the sequence diagram showed the order of communications, and the activity diagrams detailed business processes. This approach helped to identify potential problems, such as delays or suboptimal coordination, and suggest ways to eliminate them, including through automation and clear division of responsibilities. The use of UML diagrams ensures the flexibility, scalability and reliability of the system, helping to optimize logistics processes and increase the efficiency of parcel delivery and storage.

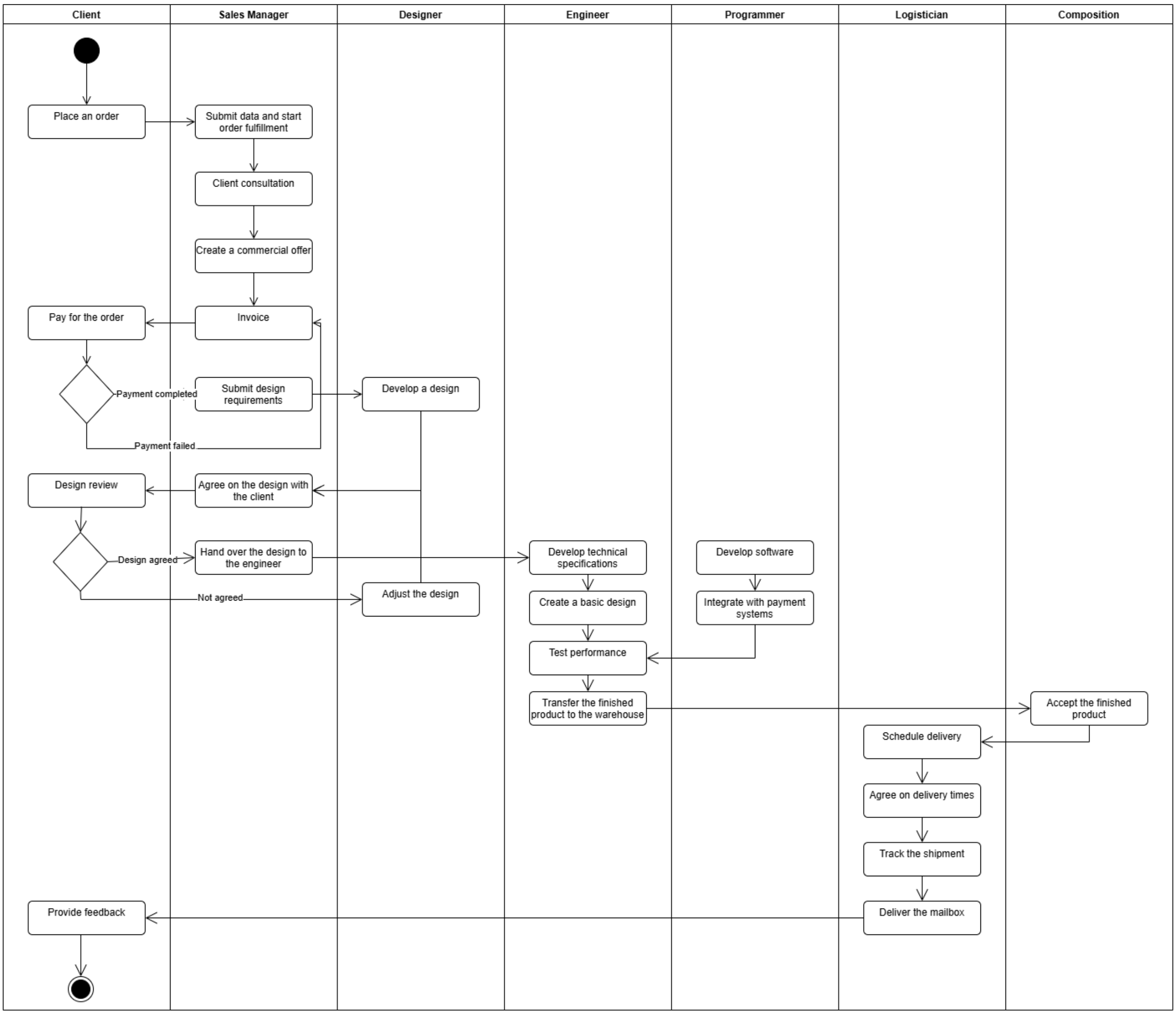

Fig. 4. UML diagram of a complex system process

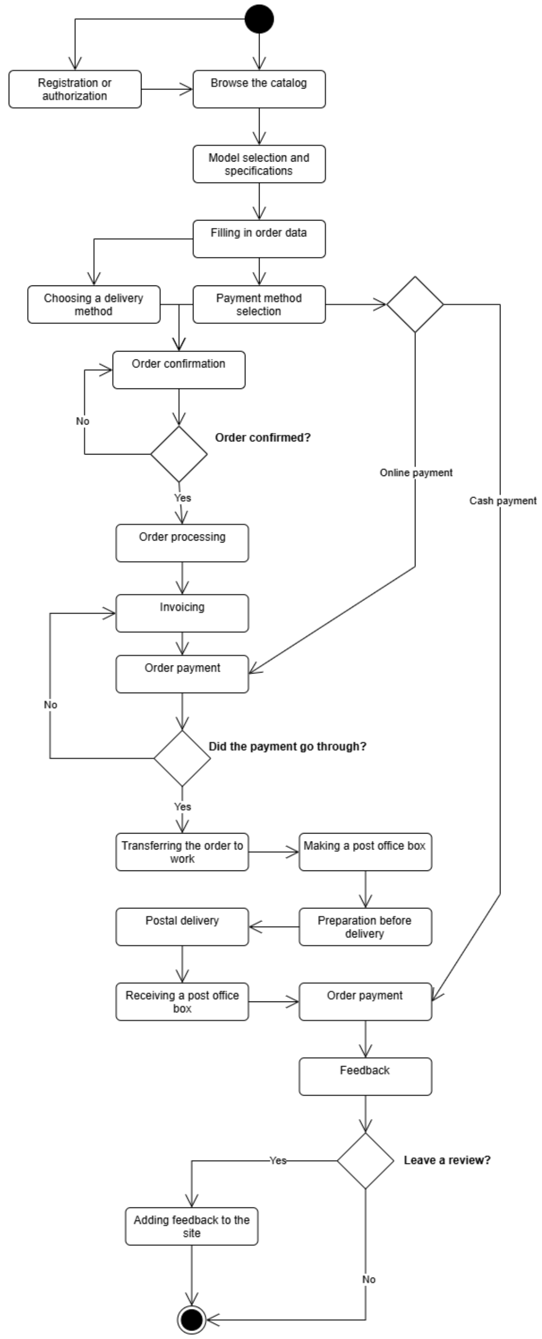

Fig. 5. UML activity flow diagram of the post office order processing process from the customer's perspective

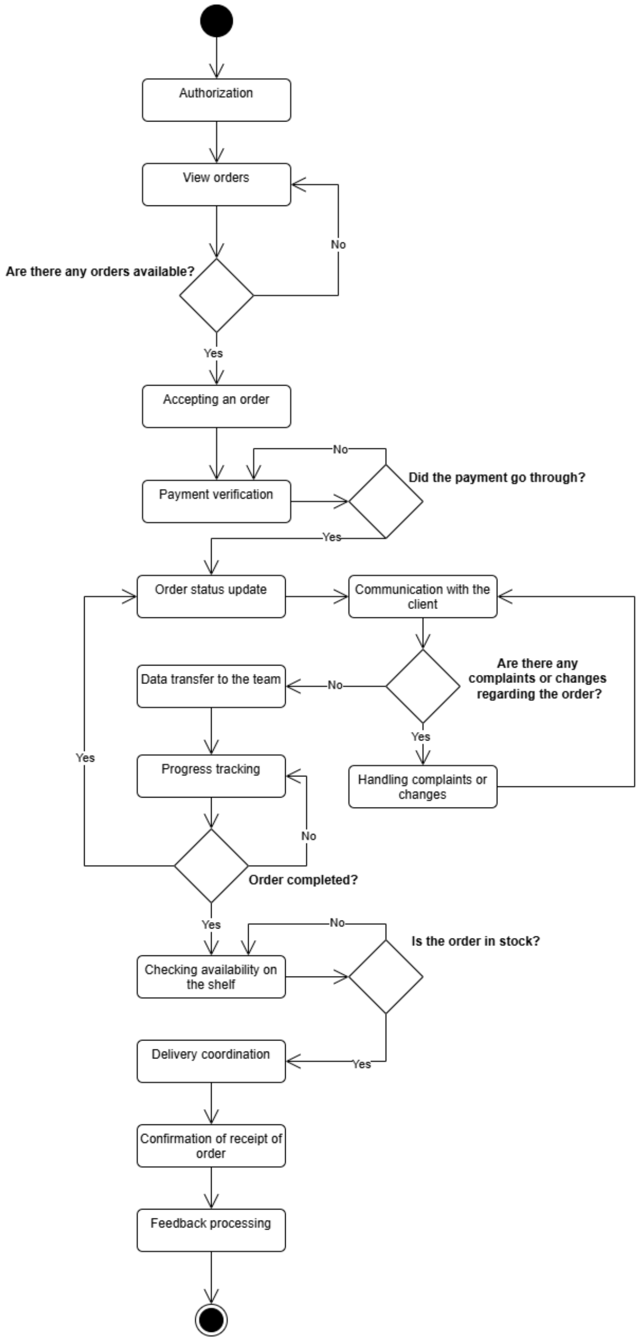

Fig. 6. UML activity diagram of the event flow process of the manager monitoring existing orders

4. Conclusions

In the course of studying the interaction of the components of a modular parcel storage system using UML diagrams, important results were achieved that allow a deeper understanding of the architecture and functioning of the system. Thanks to the use case diagram, it was possible to clearly define the main functional requirements for the system and the interaction between the user and the system, which allowed us to identify the most critical usage scenarios. Class modeling made it possible to display the data structure and organization of objects in the system in detail, which became the basis for further software development.

The activity diagram allowed us to consider the main processes in the system, in particular the processing of parcels, starting from their reception and ending with delivery. This visualized the key stages of interaction between the system components and identified possible points for optimization. The sequence diagram helped to focus on the interaction of objects when performing specific operations, which made it possible to accurately determine the order of actions and identify potential problems in communication between components.

The results obtained can become the basis for further development of the parcel storage system. In particular, it is recommended to expand the use of UML diagrams by including additional views, such as component or deployment diagrams, to reflect the infrastructure architecture of the system in more detail. For more efficient creation of UML diagrams, it is advisable to use modern CASE tools that provide high accuracy and convenience in modeling. It is also important to conduct empirical testing of the created model in real conditions, which will allow to assess the effectiveness of the interaction of system components and identify possible shortcomings.

Future research can focus on analyzing the performance of the system and identifying "bottlenecks" that may limit its efficiency, as well as on studying the scalability of the system to process larger amounts of data. In addition, it is worth exploring alternative modeling methods, such as BPMN or SysML, which can provide an even deeper understanding of business processes and automation.

Therefore, the results of the study based on UML diagrams are an important step in improving the parcel storage system, and they can be used to develop effective solutions in the field of logistics and automation of warehouse processes.

References

[1] Amelia P.: Activity Diagram in UML. Guru99 [https://www.guru99.com/uk/uml-activity-diagram.html] (accessed: 4.04.2025).

[2] Beauden J.: Analyzing the Relationship Between Implementation of UML and Software Complexity. VLDB Endowment, 2025, 1–12 [https://doi.org/10.14778/3368289.3368297].

[3] Beauden J.: An Analysis of the Significance of UML Diagrams Focused on Medium-Sized Projects. ResearchGate, 2025, 1–13.

[4] Behera K.: Development of a Parcel Management System for Logistics Companies Using Web Technologies. Master's thesis. Gangadhar Meher University, India, 2024.

[5] Cainiao. International Supply Chain [https://www.cainiao.com/global-supply-chain.html?spm=a2d524.28499374.header.7.419839c5aHhV79] (accessed: 1.04.2025).

[6] Cortés J. A. et al.: Storage System for Software Quality Metrics Associated with UML Diagrams. Journal of Positive School Psychology 6(4), 2022, 9126–9132.

[7] Iwan S., Kijewska K., Lemke J.: Analysis of Parcel Lockers' Efficiency as the Last Mile Delivery Solution – The Results of the Research in Poland. Transportation Research Procedia 12, 2016, 644–655 [https://doi.org/10.1016/j.trpro.2016.02.018].

[8] Jade M.: What Is a UML Class Diagram and the Best UML Class Diagram Creator. Mindonmap [https://www.mindonmap.com/uk/blog/what-is-uml-class-diagram/] (accessed: 3.04.2025).

[9] Margolin O.: UML for Business Modeling: Why Process Diagrams Are Needed. Evergreens [https://evergreens.com.ua/ua/articles/uml-diagrams.html] (accessed: 2.04.2025).

[10] Ooi J. Z., Tan C. C.: Smart Modular Parcel Locker System Using Internet of Things (IoT). ICSET 2021. Shah Alam, Malaysia, 2021, 66–71 [https://doi.org/10.1109/ICSET53708.2021.9612542].

[11] Parcel Pending. The Global Leader in Smart Package Locker Solutions [https://www.parcelpending.com/en-us/about/] (accessed: 28.03.2025).

[12] Rosik L.: Delivery Market: Is There an Alternative to Leaders? Ukrinform.ua [https://www.ukrinform.ua/rubric-economy/2111474-rinok-dostavki-ci-e-alternativa-lideram.html] (accessed: 25.03.2025).

[13] Samchuk L., Povstiana Y.: UML Diagrams of the Management System of Maintenance Stations. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska – IAPGOS 14(4), 2024, 141–145.

[14] Smart Locker System and Method of Parcel Delivery. United States Patent US20150186840A1, filed 23 Dec. 2014, published 2 July 2015.

[15] Studfile. Purpose and Rules for Building a UML Use Case Diagram: Types of Relationships Between Use Cases (with Examples), Representation of Roles (Actors) in a Use Case Diagram [https://studfile.net/preview/9404039/page:10/] (accessed: 2.04.2025).

[16] Zeng Y., Tong Y., Chen L.: Last-Mile Delivery Made Practical: An Efficient Route Planning Framework with Theoretical Guarantees. VLDB Endowment 13(3), 2019, 320–333 [https://doi.org/10.14778/3368289.3368297].

[17] Zosym M.: Sequence Diagrams. Maxzosim [https://www.maxzosim.com/sequence-diagrams/] (accessed: 4.04.2025).

Ph.D. Lyudmila Samchuk
e-mail: samchuk204@gmail.com

Candidate of Technical Sciences, associate professor, Faculty of Transport and Mechanical Engineering, Department of Applied Mechanics and Mechatronics, Lutsk National Technical University.

https://orcid.org/0000-0003-2516-045X

Ph.D Yuliia Povstiana
e-mail: yuliapovstyana@ukr.net

Candidate of Technical Sciences, associate professor, Faculty of Computer and Information Technologies, Department of Software Engineering, Lutsk National Technical University.

https://orcid.org/0000-0001-5426-4157

Hryshchuk Anastasia
e-mail: grisuknasta278@gmail.com

Bachelor's student at Lutsk National Technical University.

https://orcid.org/0009-0009-4889-9637