Development of a CRM System with a Decision Support Module for Engineering Component Selection

SVITLANA SOTNIK
Department of Computer-Integrated Technologies, Automation, Robotics and Safety Engineering
Kharkiv National University of Radioelectronics
Kharkiv,
UKRAINE

Abstract: - Engineering component selection in automated and industrial systems is a complex and time-consuming process due to heterogeneous technical parameters, compatibility constraints, and subjective decision-making. Existing CRM and product management systems are primarily oriented toward business processes and provide limited support for formalized engineering decision-making. This paper presents the development of a CRM system with a decision support module for engineering component selection based on the formalization of technical characteristics, compatibility rules, and selection criteria. A mathematical model is proposed to represent component attributes, requirement constraints, and compatibility relations, enabling automated filtering and ranking of feasible solutions. Based on the derived functional and non-functional requirements, a three-tier system architecture is designed to ensure performance, scalability, and maintainability. The system is implemented as a web-based solution with database-level filtering and standard CRM mechanisms, including role-based access control. Experimental evaluation shows a reduction in manual selection time by approximately 30–40 %**,** while ensuring full compliance with predefined compatibility constraints. The proposed approach improves decision repeatability and supports efficient engineering workflows in automated system design.

Key-Words: - Component selection, Engineering CRM, Architecture of the CRM, Technical decision support, Multi-criteria analysis, Compatibility verification.

Received: May 27, 2025. Revised: September 14, 2025. Accepted: December 18, 2025. Published: May 13, 2026.

1 Introduction

1.1 Relevance and Purpose of Work

In the current era of digital transformation in industry and logistics, the role of information systems [1], [2], [3] that support engineering solutions at the stages of design, implementation, and operation of automated technical systems is expanding. One of the critical tasks is the rational selection of technical components, on which the reliability, accuracy, energy efficiency, and cost of automated solutions in industrial, warehouse, and cyber-physical systems directly depend.

Within the scope of this study, technical components are understood to be sensors and actuators used in automated technical systems.

The process of engineering selection for technical components is typically based on the analysis of numerous technical parameters, compatibility standards, operating conditions, and project constraints [4]. In the absence of specialized decision support tools, this process is labor-intensive, poorly formalized, and largely dependent on the subjective experience of the engineer, which increases the risk of errors and irrational technical decisions.

Modern Customer Relationship Management (CRM) systems are traditionally used to manage customer interactions, but their architecture, which is geared toward working with large volumes of structured data, user roles, business logic, and analytics, creates the conditions for using CRM approaches in engineering support and technical component selection tasks [5], [6]. At the same time, existing CRM solutions are not adapted to the specifics of engineering tasks, in particular to the multi-criteria analysis of the technical characteristics of sensors and actuators, compatibility testing, and system reliability assessment.

This research is at the intersection of computer science, information systems, and applied engineering, covering issues of software system architecture, databases, and decision support algorithms.

The relevance of this research is due to the need to develop a specialized CRM system focused on supporting the engineering selection of automated system components. Such a system should provide centralized storage of technical data, formalization of selection algorithms, and a user-friendly interface for different categories of users. Of particular importance is the integration of algorithmic data processing methods and decision-making logic into the software architecture of the CRM system.

Insufficient automation of engineering component selection leads to increased design time, difficulties in scaling solutions, limited repeatability of results, and increased costs during system implementation and operation. In addition, the lack of a centralized knowledge base complicates the analysis of component compatibility, reliability assessment, and configuration optimization for specific application conditions.

These challenges point to the need for a software solution that combines CRM principles for managing structured data and users with powerful analytics to support engineering decisions. The main obstacle is the adaptation of the general CRM architecture to the specific requirements of technical selection, which requires expanding its functionality with modules for multi-criteria optimization, compatibility verification, and working with technical ontologies.

Thus, there is a need to develop a CRM system to support the engineering selection of sensors and actuators, combining data management, requirements formalization, and automated decision support functions. Such a system allows structuring information about technical components, reducing the influence of the human factor, and increasing the overall efficiency of the design process.

The purpose of this work is to develop a CRM system with a decision-making support module designed for the engineering selection of technical components of automated systems, which ensures increased efficiency.

The main tasks are:

1. Analysis of existing CRM solutions, identification of features.
2. Formalization of requirements for a CRM system to support the selection of technical components.
3. Development of the architecture and algorithms for the functioning of the CRM system.
4. Implementation of the software part of the CRM system.

1.2 Work-related Analysis

Modern CRM systems are widely studied as tools for managing customer data, business processes, and analytics [7], [8]. Their architectural features – the use of relational databases, role-based access, modular structure, and scalability – create the prerequisites for applying CRM approaches in technically oriented information systems. However, existing studies mainly focus on the business aspects of CRM (increasing customer loyalty, optimizing sales) and do not consider the possibility of adapting such systems to the tasks of engineering analysis and selection of technical components.

Traditional CRM systems, focused on business processes, are designed to manage customer data, automate sales and marketing communications [9], [10], [11]. Their architecture is optimized for working with structured records about customers, deals, and interaction history, and their key logic is focused on the sales funnel, reminders, and conversion analysis. In contrast, the task of engineering component selection requires working with technical parameters, compatibility standards, and ontologies of sensors and actuators, where multi-criteria optimization algorithms, compatibility checks, and reliability assessments for specific operating conditions play a key role. Therefore, the existing gaps lie in the fundamental incompatibility of data models, logic mechanisms, and performance criteria. Traditional CRMs are not adapted to processing complex technical ontologies, integrating with engineering knowledge bases, or implementing technical decision-making algorithms, which necessitates deep adaptation or the development of a specialized solution.

Despite this functional mismatch, the basic CRM architecture offers a valuable foundation for specialized engineering systems. The key concept is the management of «relationships» and entity lifecycles [12].

Among the most popular commercial CRM solutions are Salesforce [13], Microsoft Dynamics 365 [14], HubSpot [15], [16], as well as local systems for the Ukrainian market, in particular KEY CRM [17], [18] and amoCRM [19], [20].

The authors in [13] describe Salesforce as a leading customer relationship management system. In response to the growth in data volumes, Salesforce has emerged as a key tool that helps companies effectively use customer information to improve customer engagement. Such systems offer advanced customer management, marketing automation, and analytics functionality, but their focus on commercial processes makes them unsuitable for technical tasks without significant customization.

Work [13] is devoted to researching how generative artificial intelligence (GenAI) is implemented in CRM to improve its effectiveness.

Work [17] describes KEY CRM, which is used in various small and medium-sized businesses, including online stores that sell through marketplaces (Rozetka, Prom, Allo).

Works [19] and [20] describe AmoCRM as a specialized platform for customer relationship management, whose architecture and functionality are specifically optimized for automating and improving sales processes. Its main focus is on the effective execution of the sales cycle, which includes a funnel, analytics, and communication tools to maximize conversion and close deals. This positions the system as a solution that aims primarily to improve the efficiency of sales departments.

One of the well-known cloud-based customer relationship management systems is Zoho CRM [5], [21], [22]. Research [21] shows that Zoho CRM offers an integrated ecosystem for business process automation, including modules for sales, marketing, and customer support management. The study [22] describes the AI assistant Zia, which uses machine learning for data analysis and forecasting. However, like other universal CRMs, Zoho is focused on commercial processes and does not include functionality for the technical selection of components based on engineering parameters.

An alternative to the CRM approach is specialized product lifecycle management (PLM) systems, such as Siemens Teamcenter [23], PTC Windchill [24], and Dassault ENOVIA [25]. Research [23] shows that PLM systems are effective for managing BOM engineering specifications and technical documentation in post-project stages. However, these systems are designed for other purposes and users. PLM systems are aimed at design engineers, work mainly with CAD models at the detailed design stage and do not include customer relationship management functionality. For tasks involving the engineering selection of sensors and actuators at the project concept stage, with the possibility of client and manager involvement, such systems are unsuitable without significant customization and high licensing costs.

The studies in the field of CRM systems focus on the fact that each of the above CRM platforms is distinguished by its unique functionality and architectural solutions, which determine its target audience and optimal areas of application in the business environment. This diversity allows companies to choose the solution that best suits the specifics of their operations, industry requirements, and strategic goals, thereby ensuring more effective adaptation of the technology to specific business processes. However, analysis of these works shows that Salesforce, Microsoft Dynamics 365, and HubSpot, like most universal CRMs, do not have specialized modules for working with technical ontologies of sensors and executive mechanisms. Such CRMs do not contain built-in algorithms for multi-criteria analysis based on parameters such as accuracy, energy efficiency, temperature range, etc.

2 Formalization of the Problem and Requirements for the CRM System

In this work, the CRM system is considered not in the classical marketing sense, but as an information platform for managing engineering data, users, and decision-making support processes for selecting technical components of automated systems.

The functions of automatic component selection, multi-criteria analysis of technical characteristics, compatibility checking, and technical suitability assessment are implemented in the form of specialized engineering subsystems integrated into the CRM environment.

Thus, CRM acts as a basic architectural shell that provides centralized storage of structured technical data, a role-based access model, engineering process management, and decision support within the scope of component selection.

The process of selecting technical components for automated systems is a complex multi-stage task that requires simultaneous consideration of a large number of technical, operational, and design parameters. To ensure that this process can be algorithmized within the information system, it must be formalized.

Let the set of types (classes) of technical components available in the system be K:

$$\mathrm{K} = \{\mathrm{k}_1, \mathrm{k}_2, \dots, \mathrm{k}_\mathrm{m}\}\,, \tag{1}$$

where k_i – types (classes) of technical components, for example, pressure sensor, temperature sensor, actuator, etc. Here, m is the number of component types (classes).

Let us define a finite set of specific technical components available in the CRM system catalog C:

$$\mathrm{C} = \{c_1, c_2, \dots, c_n\}\,, \tag{2}$$

where c_i – specific technical components;

n – total number of components in the database.

Each component $\mathrm{c} \in \mathrm{C}$ is described by a vector of technical specifications:

$$\mathrm{h(c)} = (\, h_1(c)\,, h_2(c)\,, \dots,\, h_\mathrm{p}(\mathrm{c})\,)\,, \tag{3}$$

where $h_j(c)$ is the value of the j-th characteristic for component c;

p – the total number of characteristic types in the system.

The user query is formed as a set of requirements:

$$R = \{r_1, r_2, \dots, r_k\}, \quad (4)$$

where r_j – separate requirement. Each requirement is a logical condition or a constraint imposed on the characteristics of components;

k – number of requirements.

The set of requirements R is divided into two disjoint subsets:

- hard constraints $R_{hard} \subseteq R$ – requirements that must be satisfied mandatorily (for example, component type, operating temperature range). A component that fails to satisfy at least one requirement R_{hard} is considered unacceptable;

- soft criteria (optimization criteria) $R_{soft} = cr_1, cr_2, \dots, cr_q \subseteq R$ – requirements that define the desirable properties of a component (for example, maximizing accuracy, minimizing cost). They are used to rank the acceptable components.

Thus $R = R_{hard} \cup R_{soft}$, and $R_{hard} \cap R_{soft} = \emptyset$. The basic component selection task consists of finding all elements of set c that satisfy the system of hard constraints R_{hard}:

$$C^* = \{c \in C \mid \forall r_j \in R_{hard} : c\ satisfies\ r_j\}. \quad (5)$$

After the initial selection of individual components that meet the project's technical requirements, it becomes necessary to verify their ability to work together in a single system. The task of compatibility verification involves determining whether the selected technical components can function together to form a correct and operational technical configuration.

Let the set of components selected for a single project be:

$$C_{proj} = \{c_1, c_2, \dots, c_s\} \subseteq C, \quad (6)$$

where c_1, c_2 – are the components of the project; s is the number of components in the project.

To formalize compatibility, let us introduce a binary compatibility relation:

$$BCR \subseteq C \times C, \quad (7)$$

it is a subset of the Cartesian product of the set of components with itself.

The pair $(c_i, c_j) \in BCR$ means that component c_i is technically compatible with component c_j when used together.

The relation can be asymmetric: the fact that a pair $(c_i, c_j) \in BCR$ (i.e., component c_i is compatible with c_j) does not necessarily imply that the pair (c_i, c_j) also belongs to BCR. For example, this applies to a relationship of the type «the output of c_i connects to the input of c_j» – the reverse connection may be technically impossible.

A configuration C_{proj} is considered technically compatible if and only if the compatibility condition is satisfied for every pair of distinct components within the project:

$$\forall c_i, c_j \in C_{proj}, i \neq j : (c_i, c_j) \in BCR. \quad (8)$$

In a real system, the BCR ratio is not explicitly stored for every possible pair of components (which would lead to a quadratic increase in data volume). Instead, it is determined dynamically based on a set of technical rules and component attributes.

Compatibility rules can be of the following types:

1. Interface compatibility: Output interface type $(c_i) \in$ List of supported inputs (c_j).

2. Electrical compatibility: Operating voltage (c_i) = Supply voltage (c_j) or Current consumption $(c_i) \leq$ Maximum output current (c_j).

3. Protocol compatibility: Communication protocol version(c_i) = Protocol version(c_j).

4. Mechanical/Spatial compatibility: Fastening type (c_i) = Fastening type (c_j) (for certain classes of components).

The verification of compatibility for a component set C_{proj} can be represented as a Constraint Satisfaction Problem (CSP). Let us define:

- a set of variables: $X = \{x_1, x_2, \dots, x_s\}$, where each variable x_i corresponds to a component $c_i \in C_{proj}$;

- domains of the variables: D_i – the set of allowable values for variable x_i, containing all characteristics of component c_i that affect compatibility (interface type, voltage, protocol, etc.);

- a set of constraints: Constraints – a collection of predicates defined on subsets of variables. Each constraint encodes a compatibility rule. For example, for a pair of components (x_a, x_b) that must be connected, the constraint takes the form:

$$\text{InterfaceType}(x_a.\text{output}) \in$$

$$\in \text{SupportedInterfaces}(x_b.\text{input}).$$

Configuration C_{proj} is considered compatible if and only if there exists an assignment of values to the variables from their respective domains such that all constraints in the set Constraints are satisfied. This constitutes a formalization of the binary compatibility relation BCR (formula 7) via a computable CSP model.

The practical implementation of compatibility verification is carried out by applying rules of types 1-4 (interface, electrical, protocol, and mechanical/spatial compatibility) to all pairs of components in the project configuration.

If the set of feasible components C^* resulting from the basic selection turns out to be too large for effective analysis, a mechanism is required to determine the best options. The ranking task involves ordering the elements of set C^* according to their degree of compliance with the original project requirements.

Let $q = |R_{soft}|$ be the number of soft criteria. For each soft criterion $cr_j \in R_{soft}$, a fitness (or utility) function is defined:

$$f_j(c) : C \rightarrow \mathrm{R}, \tag{9}$$

which evaluates how well component c satisfies the j-th criterion.

For practical implementation, a scalarization method using a weighted sum is applied, which transforms the multi-criterion problem into a single-criterion one. A user or expert assigns weight coefficients w_j for each criterion cr_j, where:

$$\sum_{j=1}^{q} w_j = 1, w_j \geq 0. \tag{10}$$

For each component $c_i \in C^*$, an integral fitness score is calculated:

$$F(c_i, R) = \sum_{j=1}^{q} [w_j \cdot v_j(c_i)], \tag{11}$$

where $v_j(c_i)$ is the normalized value of component c_i for criterion cr_j.

Normalization $v_j(c_i)$ is performed using a three-level scale based on the fitness function $f_j(c_i)$ (12):

$$v_j(c_i) = \begin{cases} 1, if\ f_j(c_i) \text{ fully satisfies the requirement,} \\ 0{,}5,\ \ if\ f_j(c_i) \text{ partially satisfies the requirement,} \\ 0, if\ f_j(c_i) \text{ does not satisfies the requirement.} \end{cases}$$

The result of solving the ranking problem is a list sorted in descending order of $F(c_i,R)$:

$$C_{ranked} = sort\ (C^*, F, descending)\ , \tag{13}$$

the top positions are occupied by technical components with the maximum fitness score.

Thus, the formal task of engineering component selection is transformed into a multi-criteria optimization problem with decision support (Multi-Criteria Decision Support), which combines:

- hard constraints (R_{hard}) → basic selection (formulas 4-5);
- compatibility verification (BCR) → CSP model (formulas 6-8);
- soft criteria (R_{soft}) → ranking (formulas 9-13).

The system not only filters out unacceptable options but also provides the engineer with well-founded recommendations for optimal component selection based on formalized criteria and automated compatibility verification.

3 Development of CRM System for Engineering Component Selection

3.1 Functional and Non-Functional Requirements Analysis

Developing an effective CRM system for engineering recruitment requires a clear definition of its requirements in the context of the identified shortcomings of existing approaches. To formalize these requirements, a methodology based on the ISO/IEC/IEEE 29148:2018 standard was applied, which provides for a clear division into functional (FR) and non-functional (NFR) requirements. Key requirements have been formulated to directly address the problems described in the introduction: labor intensity, subjectivity, and lack of formalization.

Functional requirements (FR) directly derived from the formal selection task:

FR1. Advanced search and multi-criteria filtering of components. The system shall provide an intuitive interface for forming requests R and implementing the basic selection task (formulas 4-5). The search shall support filtering of sensors and actuators by technical characteristics from the characteristics vector h(c) (for example, measurement range, accuracy, output signal type, communication interface), operating conditions, and manufacturer. This will eliminate the labor-intensive manual analysis of catalogs and ensure the efficient formation of the set C^* of admissible components.

FR2. Project Management and Selection History. The system shall automatically save the history of user queries and selected components, with the capability to create, save, and reuse typical configurations C_{proj}. This facilitates process formalization, knowledge accumulation, and efficient work with complex technical solutions. Each project stores the structure of selected components along with their quantity through an intermediate relationship model.

FR3. Automated Component Compatibility Verification. To implement the compatibility verification task (formulas 6-8), the system must contain a mechanism for automatic or semi-automatic verification of selected sensors and actuators for compatibility based on electrical, interface, and software parameters. This means the practical implementation of the binary compatibility relation BCR through a computable CSP model using rules of types 1-4 (interface, electrical, protocol, and mechanical compatibility). Automation of verification reduces subjectivity and the risk of errors when forming technical configurations.

FR4. Ranking and Decision Support. The system shall implement a ranking algorithm based on the weighted sum (formulas 10-13), which formalizes the fitness function $F(c_i, R)$. A user or expert assigns weight coefficients w_j for each soft criterion $cr_j \in R_{soft}$ (e.g., accuracy, cost, energy efficiency), where $\sum_{j=1}^{q} w_j = 1$.

For each component $c_i \in C^*$, an integral score is calculated using formula (11), where the normalized value $v_j(c_i)$ is determined according to the three-level scale (see Section 2, formula 12).

The result is a list C_{ranked} (formula 13), sorted in descending order of $F(c_i, R)$. The system should also provide the capability for interactive adjustment of criterion weights with instant recalculation of ratings and visualization of results in the form of tables and charts.

Among the key non-functional requirements (NFR), the following should be highlighted:

NFR1. Usability. The system interface must be intuitive for design engineers, requiring minimal training. The role-based access model should ensure interface adaptation to specific usage scenarios (client, manager, administrator) without excessive information.

NFR2. Performance. The system's response time for standard search and filtering operations should not exceed 2-3 seconds, even when working with a catalog of tens of thousands of components. The calculation of the fitness function $F(c_i, R)$ and the formation of the list C_{ranked} must be performed in real-time to ensure interactivity when changing the weight coefficients.

NFR3. Scalability. The system architecture must allow for scaling of the component database (increasing n in formula 2) and user accounts without significant performance degradation. The three-tier architectural model should enable horizontal scaling at the business logic and data storage levels.

Thus, the proposed CRM system is designed to become a key tool that transforms the complex and informal process of engineering component selection into an efficient, structured procedure. The implementation of the formal model (formulas 1-13) through the functional requirements FR1-FR4 will not only reduce design time but also improve the technical and economic indicators of the automated systems themselves (reliability, accuracy, cost), thereby enhancing the competitive advantages of enterprises in the digital economy.

3.2 Development of the architecture of the CRM system

In the process of developing a CRM system to support the engineering selection of technical components, a key stage is the formation of its software architecture, capable of ensuring effective work with large volumes of structured technical data, the implementation of decision support algorithms, and the interaction of different categories of users. The system architecture must meet the requirements of scalability, modularity, component reuse, maintainability, and information security, which are critical for modern engineering information systems.

Taking these requirements into account, a three-tier architectural model was chosen, which provides a clear logical distinction between the presentation layer, the business logic layer, and the data layer (Figure 1).

This approach is common in enterprise-level systems and allows interface solutions to be isolated from computational logic and physical data storage, which simplifies further development and modification of the system.

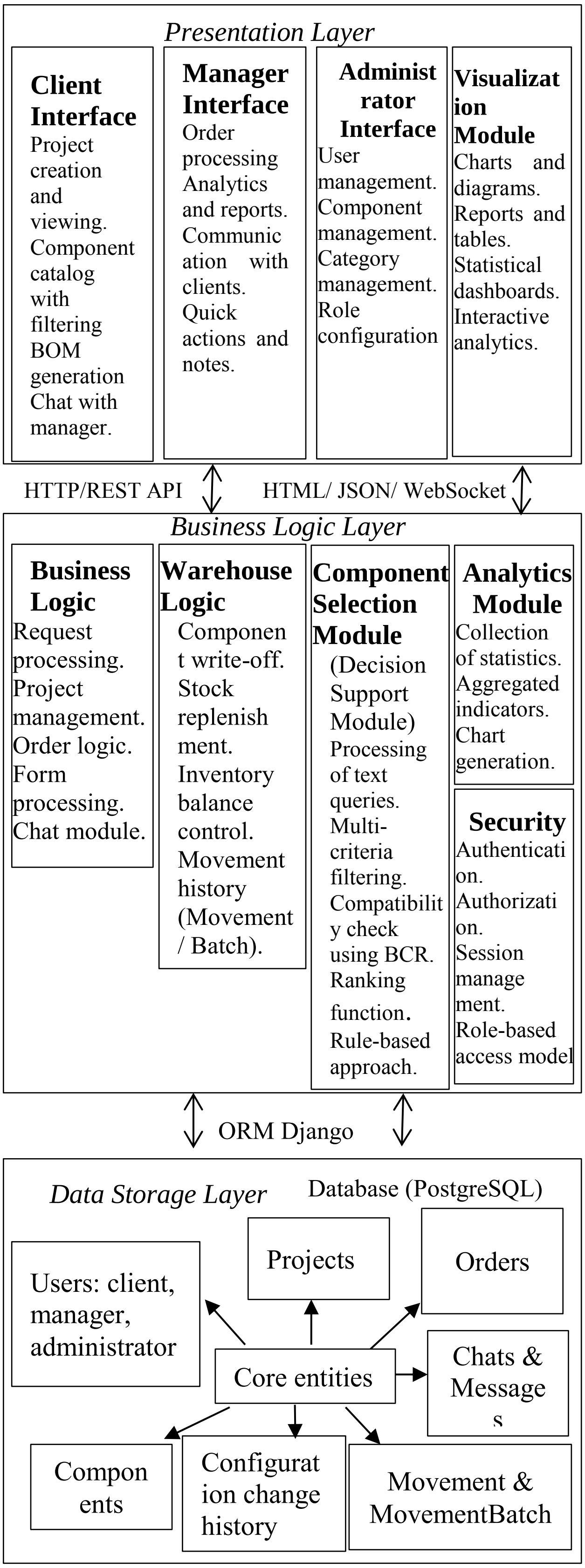


Fig. 1: Architecture of the CRM system for engineering selection of technical components

Source: created by the authors.

The presentation layer is responsible for user interaction with the CRM system and implements the interface for accessing its functional capabilities. The proposed architecture provides separate interfaces for customers, managers, and administrators, allowing for the implementation of a role-based access model and the adaptation of functionality to the tasks of each user category.

The client interface allows you to create and view engineering projects, work with a catalog of sensors and actuators with support for multi-criteria filtering, and generate queries to the decision support system. The manager interface is focused on order support, analysis of technical configurations, working with statistical data, and communication with clients. The administrative interface implements functions for managing users, the component base, categories, and access policies.

A separate visualization module is used to visualize the results of engineering analysis and analytical calculations, generating graphs, tables, and summary reports. Data exchange between the presentation layer and the server side is carried out using standard web protocols (HTTP/HTTPS) and JSON data transfer formats, as well as WebSocket connections for implementing interactive scenarios.

The presentation layer is responsible for user interaction with the CRM system and implements the interface for accessing its functional capabilities. The proposed architecture provides separate interfaces for customers, managers, and administrators, allowing for the implementation of a role-based access model and the adaptation of functionality to the tasks of each user category.

The client interface allows you to create and view engineering projects, work with a catalog of sensors and actuators with support for multi-criteria filtering, and generate queries to the decision support system. The manager interface is focused on order support, analysis of technical configurations, working with statistical data, and communication with customers. The administrative interface implements functions for managing users, the component base, categories, and access policies.

A separate visualization module is used to visualize the results of engineering analysis and analytical calculations, generating graphs, tables, and summary reports. Data exchange between the presentation layer and the server side is carried out using standard web protocols (HTTP/HTTPS) and JSON data transfer formats, as well as WebSocket connections for implementing interactive scenarios.

The business logic level is the central element of the CRM system and implements functionality related to processing user requests, data management, and supporting engineering decisions. It is at this level that algorithms for multi-criteria selection of sensors and actuators, component compatibility checks, and analysis of their technical characteristics are concentrated.

The business logic of the system includes modules for processing engineering requests, project management, inventory accounting, and generating analytical indicators. The analytical module collects statistical data on component usage, generates aggregate indicators, and builds diagrams, allowing for the evaluation of technical solutions' effectiveness.

A separate component of the architecture is the Component Selection Module, which is a software implementation of the mathematical decision support model formalized in Section 2 (formulas 1-11). The module is implemented at the business logic level using a rule-based approach. This module provides processing of user text queries through tokenization and extraction of key parameters using regular expressions, multi-criteria filtering of components through SQL queries to the database, as well as automated verification of the compatibility of selected sensors and executive mechanisms based on specified technical criteria.

Information security is ensured through authentication, authorization, session management, and protection against typical web threats. Placing security functions at the business logic level allows for centralized control of access to data and system functionality.

The data storage level is designed for centralized storage of information about users, technical components, projects, orders, and engineering analysis results. The PostgreSQL relational database management system is used as the basis for implementing this level, which ensures reliability, support for complex queries, and data integrity.

The database structures information about sensors and actuators, taking into account their technical characteristics, categories, subcategories, and compatibility parameters. In addition, data on customer projects, configuration change history, component selection results, and analytical indicators are stored. This organization of data creates a basis for reusing accumulated knowledge and improving the soundness of engineering decisions.

The proposed three-tier architecture directly addresses NFR2 (performance) through database-level filtering and NFR3 (scalability) by decoupling presentation, logic, and data layers.

3.3 Core Algorithms and System Implementation

The CRM system is implemented using web technologies. The backend is built on the Django framework, the frontend is based on standard web development technologies (HTML, CSS, JavaScript) using built-in Django mechanisms for templates, and the database is PostgreSQL. The automated component selection module is implemented as a rule-based algorithmic system using text query processing mechanisms (tokenization, parameter extraction via regular expressions) and multi-criteria component filtering via SQL queries to the database.

Based on the architectural solutions outlined in the previous section and formalized requirements, key algorithms for the functioning of the CRM system were developed. These algorithms cover the entire cycle of user interaction with the system, from authentication and working with the component catalog to placing and processing orders. Each algorithm is presented in the form of a flowchart with a detailed explanation of its logic.

The effective functioning of a CRM system in the context of engineering recruitment requires a clearly defined sequence of actions for each type of user. The fundamental element that ensures security and personalization is the authentication and authorization module (Figure 2). This module is responsible for verifying the user's identity and granting them access to the system's functionalities and data in accordance with their assigned role.

The algorithm begins with loading the login page. The user selects an action: if they have an account, they enter their username and password; if not, they register and are automatically assigned the role of «client».

When the credentials are entered, the system verifies their correctness. If there is an error, a message is displayed, and the user is returned to the login form.

Upon successful authentication, the system determines the user's role and redirects them to the appropriate dashboard: customer or manager.

The algorithm for user interaction with the component catalog (manual component selection) (Figure 3) implements the process of incremental (step-by-step) formation of the technical configuration of the project by sequentially selecting and adding sensors and actuators.

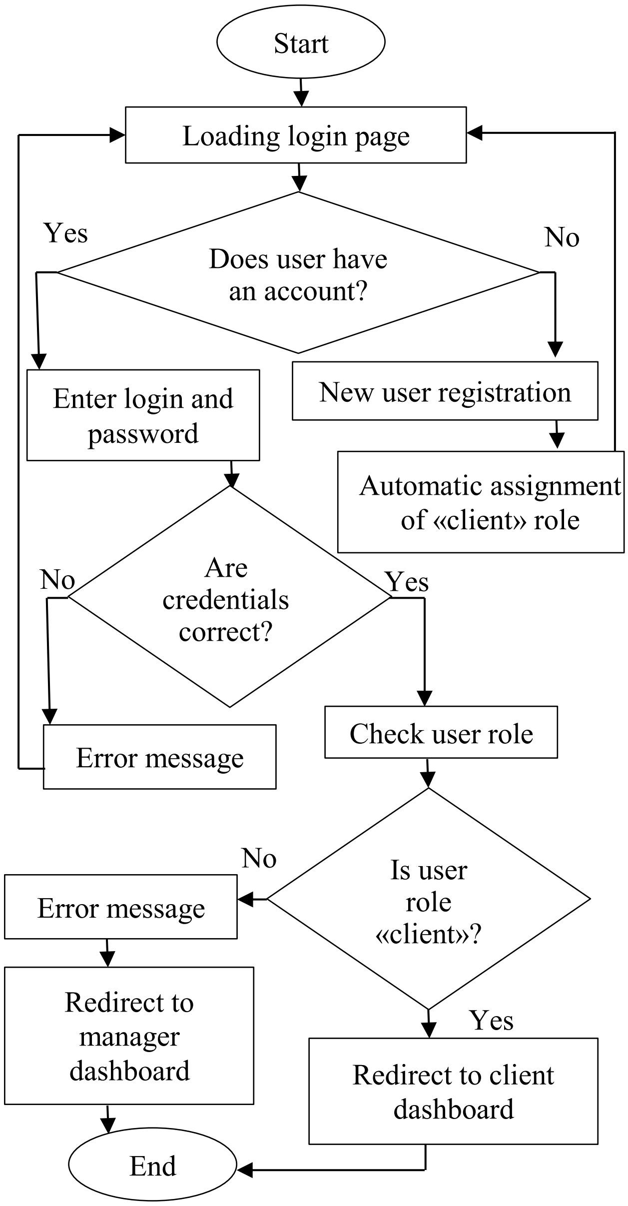


Fig. 2: Algorithm for user authorization and registration in the CRM system

Source: created by the authors.

The algorithm begins with opening the component catalog, where the user is given access to the full range of technical elements of the system.

At the next stage, the customer uses navigation tools: filtering by categories, subcategories, and brands, or performs a text search by the name of a specific component. This ensures the quick finding of the necessary items among a large number of positions.

After applying filters or searching, the system displays relevant results. The user selects a component and proceeds to view its detailed card, which contains technical specifications, images, and other related information.

Having decided to add a component, the customer initiates the appropriate action. At this stage, the system checks for existing projects in the user's account.

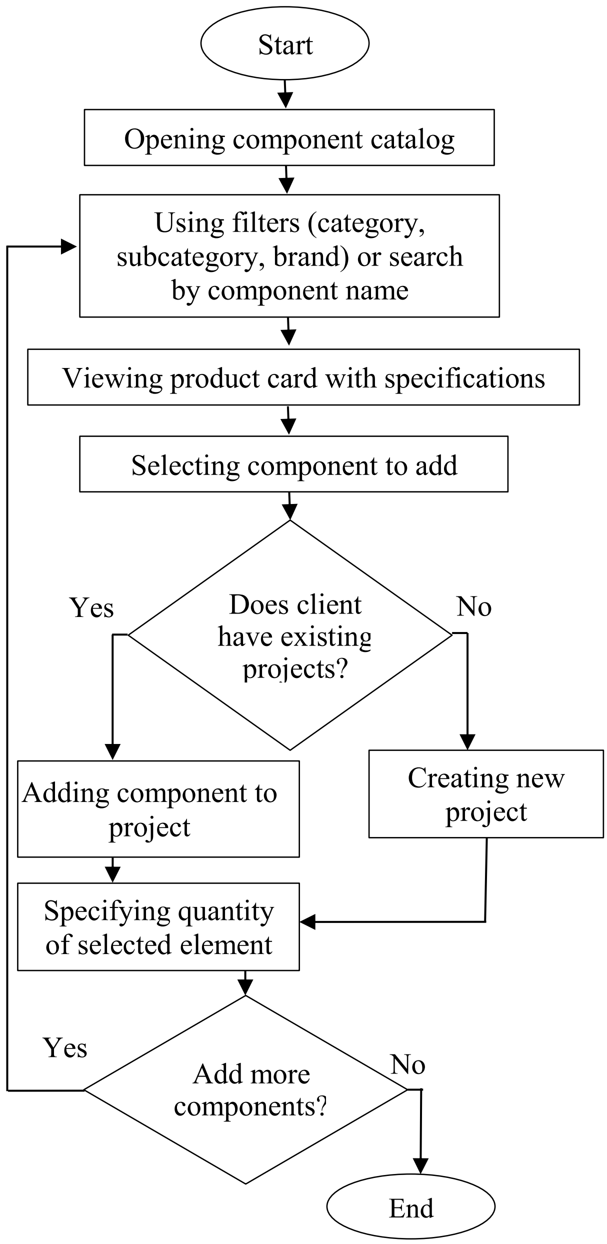


Fig. 3: Algorithm for selecting components from the catalog and forming the technical composition of the project (manual selection of components)

Source: created by the authors.

If projects already exist, the system offers to select one of them to add the component. If there are no projects, the process of creating a new project linked to the current user is automatically launched. After selecting or creating a project, the selected component is added to its technical composition. The user specifies the required number of items, which is important for further planning and analytics. After adding one technical component, the system asks the user if they want to continue working with the catalog. If so, the process returns to the filter and search stage, allowing other components to be added to the same project. If not, the algorithm ends, and the generated project is saved in the system for further order placement or consultation with a manager.One of the key elements of the developed CRM system is the component selection algorithm, which implements decision support functions through the formalization and automation of the process of selecting sensors and actuators in accordance with the specified project requirements. This algorithm implements the logic of multi-criteria analysis based on rules and reduces the influence of subjective factors in technical decision-making (Figure 4).

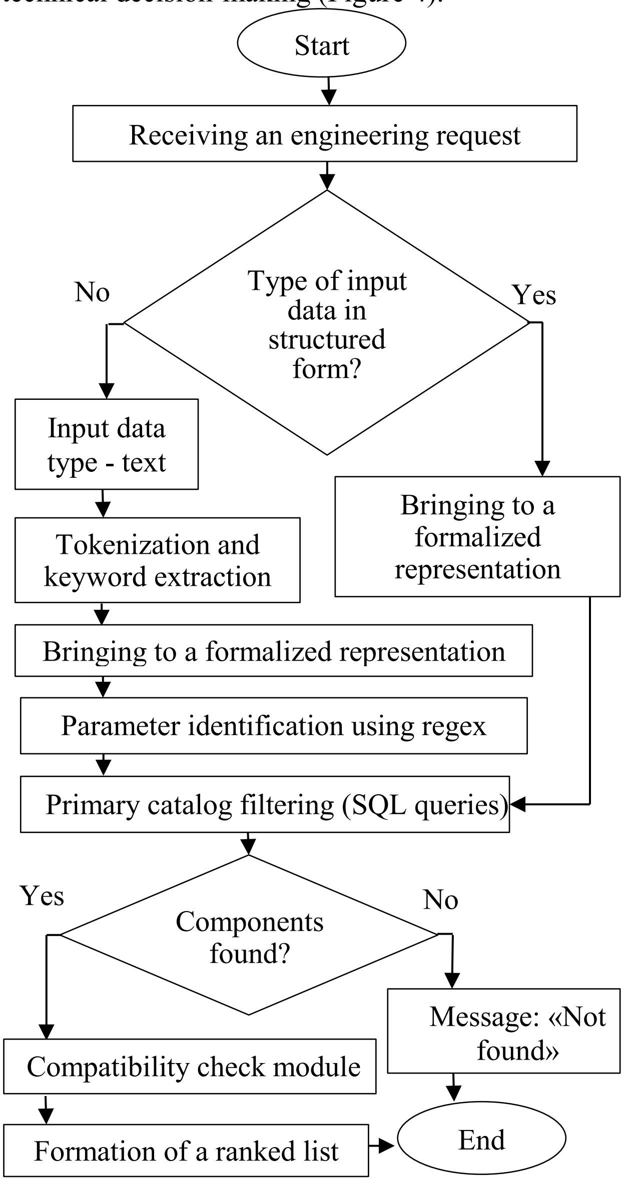


Fig. 4: Algorithm for automated component selection with decision support

Source: created by the authors.

The algorithm begins with the formation of the user's engineering request. Requirements can be specified in a structured form (using a form with technical parameters) or in free text format. For text requests, the system performs tokenization and keyword extraction using a dictionary of technical terms. Next, regular expressions are applied to identify numerical parameters (measurement ranges, voltages, frequencies) and their relationship to the corresponding component characteristics. At this stage, the input data is normalized and converted to an internal formalized representation.

In the next stage, the system performs initial filtering of components through SQL queries to the PostgreSQL database according to mandatory criteria such as component type, operating parameter range, operating conditions, and availability in stock. This significantly reduces the search space and increases the performance of the algorithm.

The algorithm for automated component selection will be described in more detail in subsequent works.

The algorithm for automated component selection will be described in more detail in subsequent works.

Next, a compatibility check module is implemented, which analyzes the electrical, interface, and logical parameters of the selected components. Incompatible options are automatically filtered out, reducing the risk of incorrect configurations in the early stages of design. The final stage involves forming a ranked list of components based on the degree of compliance with the specified criteria.

The stage of forming a ranked list of components, presented in a general form in Figure 4, is implemented by a separate evaluation and sorting algorithm. The detailed structure of this stage, including the calculation of the integral fitness function $\mathrm{F}(\mathrm{c}_i, R)$ and the ordering of the set of admissible components C^*, is shown in Figure 5.

The algorithm for forming a ranked list of technical components, presented in Figure 5, implements the stage of evaluating and ordering feasible alternatives obtained after preliminary filtering and compatibility verification. The input to the algorithm is the set of compatible components C^*, the system of optimization criteria R_{soft}, and the corresponding weight coefficients w_j, which reflect the relative importance of each criterion in the decision-making process.

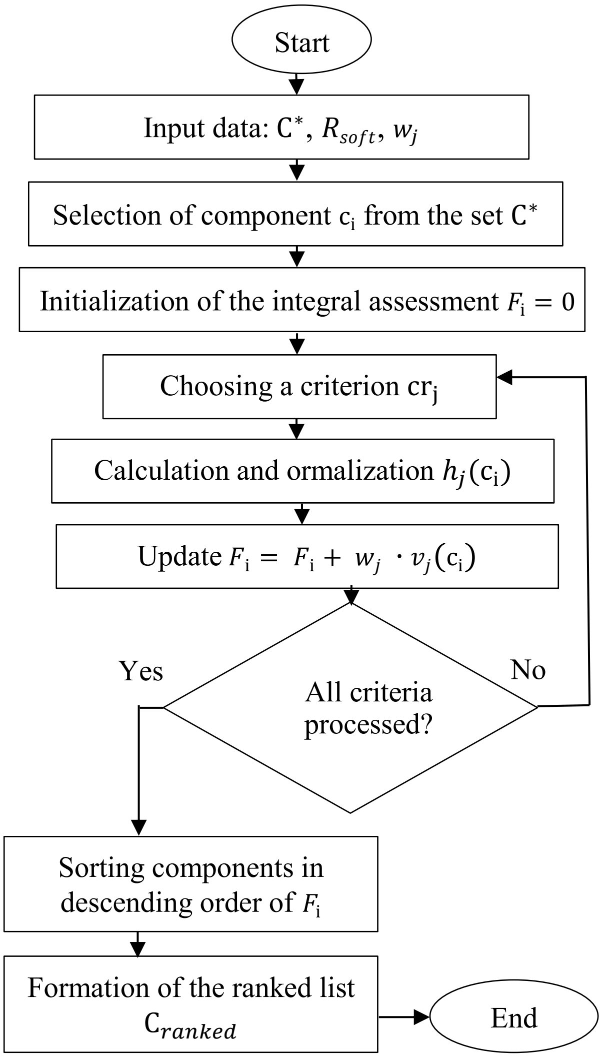


Fig. 5: Algorithm for forming a ranked list of technical components based on the fitness function $F(c_i, R)$, which implements the «Formation of a ranked list» stage

Source: created by the authors.

After algorithm initialization, a sequential selection of each component c_i from the set C^* is performed. For the current component, the initial value of the integral suitability score $F_i = 0$ is set, after which an internal evaluation loop over all optimization criteria is launched. Within this loop, each criterion $cr_j \in R_{soft}$ is sequentially selected, and the corresponding actual value of the component's technical characteristic $h_j(c_i)$ is determined.

The obtained value is normalized using the given utility function, resulting in a dimensionless score $v_j(c_i)$ that reflects the degree of satisfaction of the corresponding criterion. The normalized score is multiplied by the weight coefficient w_j, and the weighted contribution is added to the current value of the component's integral fitness function. This process is repeated until all criteria from the set R_{soft} have been processed.

After completing the evaluation for all criteria for the current component, its final integral score F_i is obtained and saved for further analysis. A similar procedure is performed for each component from the set C^*, resulting in a set of alternatives with corresponding numerical suitability scores.

At the final stage of the algorithm, the components are sorted in descending order of the values of the integral fitness function F_i. This allows the alternatives to be ordered according to their degree of compliance with the specified project criteria. The result of the algorithm's operation is a ranked list of technical components C_{ranked}, in which the top positions are occupied by components with the highest generalized quality score. The formed list is used by the decision support system to provide the engineer with substantiated recommendations for selecting optimal technical components.

Let us describe the process of analyzing the computational complexity of the ranking algorithm.

Let $N = |C^*|$ be the number of component candidates after filtering, and $Q = |R_{soft}|$ be the number of soft optimization criteria. Computing the integral score $F(c_i, R)$ for one component requires evaluation over all Q criteria, which constitutes $O(Q)$ operations. Since this procedure is performed for all N components, the overall complexity of the scoring stage is $O(N \cdot Q)$. The next stage – sorting N elements – has a complexity of $O(N \log N)$ when using efficient algorithms (e.g., quicksort). Thus, the total computational complexity of the ranking algorithm is $O(N \cdot Q + N \log N)$. In practical engineering selection scenarios, the number of criteria Q is small and fixed (typically 5–10), so the dominant term in the complexity is $O(N \log N)$. This ensures high performance of the algorithm even when working with catalogs containing tens of thousands of components and directly satisfies the non-functional requirement NFR2 (Performance) specified in section 3.1.

To illustrate the practical efficiency, let's consider a specific example.

Suppose after filtering, we obtain $N = 500$ candidate components, and the user has specified $Q = 7$ optimization criteria (accuracy, cost, power consumption, dimensions, temperature range, response time, reliability). Calculating the integral scores $F(c_i, R)$ for all components requires $500 \times 7 = 3{,}500$ operations (computing the normalized values v_j and their weighted summation). The subsequent sorting step requires approximately $500 \times \log_2(500) \approx 4{,}483$ comparison operations when using the quicksort algorithm. According to profiling results on the test server (Intel Xeon E5-2680, 16 GB RAM, PostgreSQL 14), the complete computation and formation cycle of the ranked list C_{ranked} is executed in approximately 0,7 seconds. This fully satisfies the NFR2 requirement of a response time of less than 3 seconds and enables interactive operation when adjusting the weight coefficients w_j.

In addition to automated selection tools, an important aspect of the CRM system is well-established communication between users. To this end, the system implements a chat algorithm (Figure 6) that enables direct messaging between the client and manager within the framework of a specific project or order.

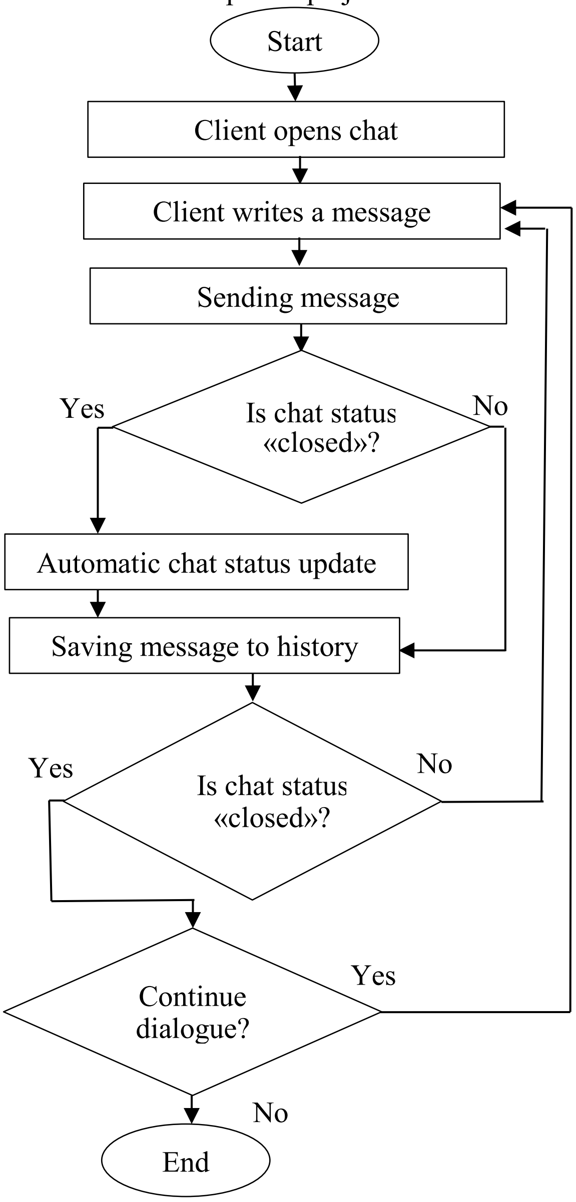


Fig. 6: Algorithm for chat between client and manager in CRM system

Source: created by the authors.

The algorithm begins when the client opens the chat tab via the system's navigation menu. This initiates the loading of the communication interface, where the user can create a new dialogue or continue an existing one.

The client formulates and writes their message in the text input field. After finishing writing, the user initiates the sending of the message by pressing the corresponding button.

At the sending stage, the system checks the current status of the dialogue. If the chat was previously closed (for example, after the previous interaction ended), the system automatically updates its status, switching the dialogue to active communication mode. This makes it possible to resume communication without having to create a new chat.

After updating the status or confirming the active state, the message is saved in the correspondence history. The entire history of the message exchange is recorded in the database, which guarantees the preservation of the dialogue context and provides the ability to review it later.

The saved message becomes available for viewing by the manager. The system waits for a response from the company representative. If the manager responds, the interaction continues, and the system checks again whether the customer wishes to continue the dialogue.

If the customer has additional questions or needs clarification, they can write a new message, and the process returns to the message-writing stage. This creates a cyclical dialogue structure that supports multiple exchanges of information.

If the customer does not wish to continue the conversation or has received a comprehensive response, the dialogue can be terminated. At the same time, the entire history of the message exchange remains stored in the system, allowing you to quickly return to previous requests if necessary.

An important feature of the algorithm is the automatic sorting of chats by date of last activity, which simplifies the manager's work with a large number of inquiries and ensures that current dialogues are processed first.

After discussing the technical details and agreeing on the configuration with the manager via chat, the customer proceeds to the final stage – placing an order. The algorithm of this process (Figure 7) formalises the user's actions in converting a technical project into a commercial order.

The algorithm begins with the user navigating to the «My Projects» tab via the system navigation menu.

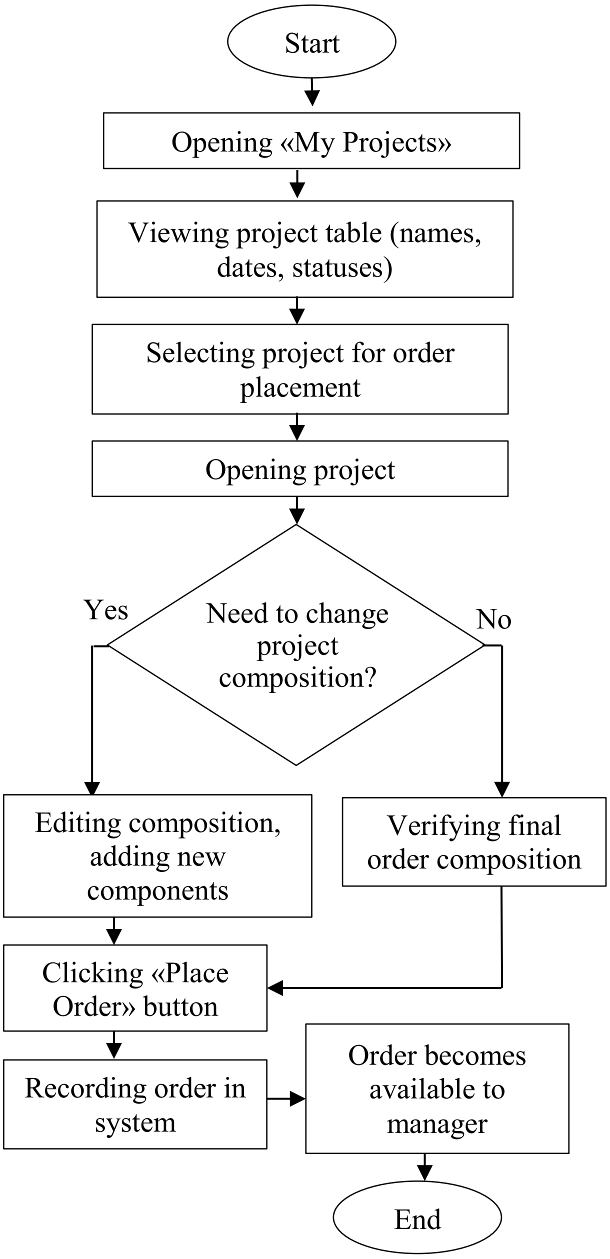


Fig. 7: Algorithm for placing an order by a customer in the CRM system

Source: created by the authors.

The screen displays a table containing a list of all projects created by the user, indicating their names, creation dates, and current statuses. This table allows you to quickly navigate among existing projects.

The user selects one of the projects that they plan to place an order for and initiates its opening.

After opening the project, the system displays its complete structure, including a list of added components, their quantity, and technical characteristics. At this stage, the user decides whether changes need to be made to the project.

If you need to change the composition, you go to edit mode, where you can add new technical components, delete existing ones, or adjust their quantity. After you finish editing, the system moves on to the next stage.

If no changes are needed, the user immediately proceeds to the final order review stage, where they can view the final configuration before placing the order.

After confirming that the order is correct, the user clicks the «Place order» button.

The system records the order in the database, assigning it the appropriate status and storing all information about the composition, number of components, and customer data.

From this moment on, the order becomes available for viewing by the manager, who can start processing it, assign themselves as responsible, and continue interacting with the customer through other channels of the system.

The algorithm ends after the order is successfully recorded and transferred to the active orders section.

After the customer places an order, it is sent to the processing system, where responsibility for its execution is transferred to the manager. The order processing algorithm by the manager (Figure 8) implements centralized logic for managing request statuses and provides control at all stages of execution.

The algorithm begins with the manager going to the «Sales» section, where a table with all orders available in the system is displayed.

The manager performs a search or applies filters to find the necessary orders. The table functionality supports filtering by status, customer names, dates, and other parameters for quick navigation.

After finding the required order, the manager selects it from the list for further processing.

The system checks whether a manager has been assigned to this order. If a manager has already been assigned, the manager returns to the search or filter to find another order. If no manager has been assigned, the manager can assign themselves to this order, and the changes are recorded in the system.

After assigning a manager, the system checks the validity of the order. If the order is invalid or the customer has canceled it, the manager cancels the order, after which it is removed from active processing.

If the order is valid, the manager proceeds to a detailed review, where the complete order composition is displayed with a list of components and their quantities.

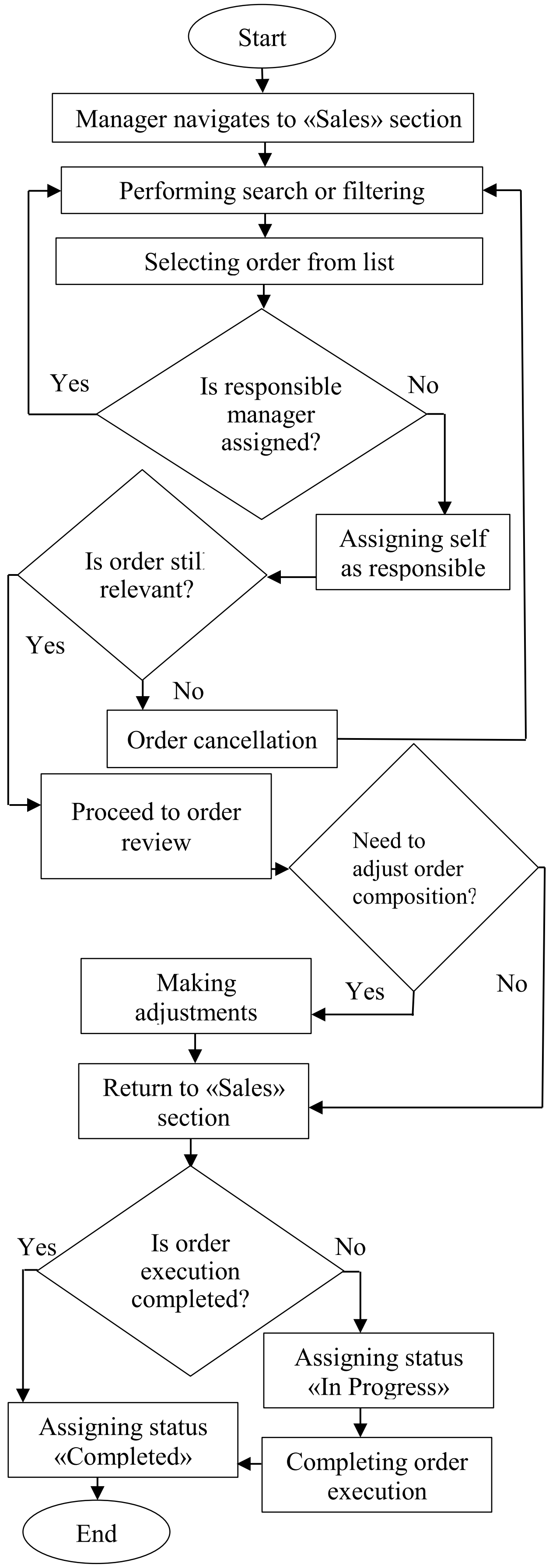


Fig. 8: Algorithm for processing user orders by the manager

Source: created by the authors.

At this stage, the manager assesses whether the order composition needs to be adjusted. If changes are necessary, he makes adjustments – changes the quantity of components, adds new ones, or deletes existing items. All changes are automatically saved in the system.

After completing the adjustments or if no changes are needed, the manager returns to the «Sales» section, where they decide on the further status of the order.

The system asks whether the order fulfillment is complete. If the order still requires processing or awaits additional actions, the manager assigns it the status «In Progress». This status indicates that the order is in the process of active handling.

If all necessary actions are completed, the components are selected, and the order is ready for shipment or transfer to the warehouse, the manager assigns it the status «Completed».

After the status is set to «Completed», the order execution is finalized. The order moves to the completed section and becomes available for further work in the warehouse module, where the components will be written off.

The algorithm ends after the status is successfully changed and all changes are recorded in the system database.

Once the manager has finished processing the order and changed its status to «Completed», the system automatically initiates an update of the warehouse inventory. The algorithm for this process (Figure 9) ensures synchronization between commercial operations and the accounting of material assets, automatically writing off components from the warehouse.

The algorithm is activated at the moment the order is assigned the status «Completed». This status serves as a trigger to start the process of updating stock levels. After the «Completed» status is set, the system automatically initiates the procedure of deducting components from the current warehouse inventory. All technical components included in the completed order are subject to subtraction from the total quantity of items available in stock.

Simultaneously with the write-off, the system creates a record of component movement of the «Write-off» type. This record logs information in the database about the volume of written-off components, their nomenclature, and the transaction date, ensuring transparency and the ability to track all warehouse stock operations.

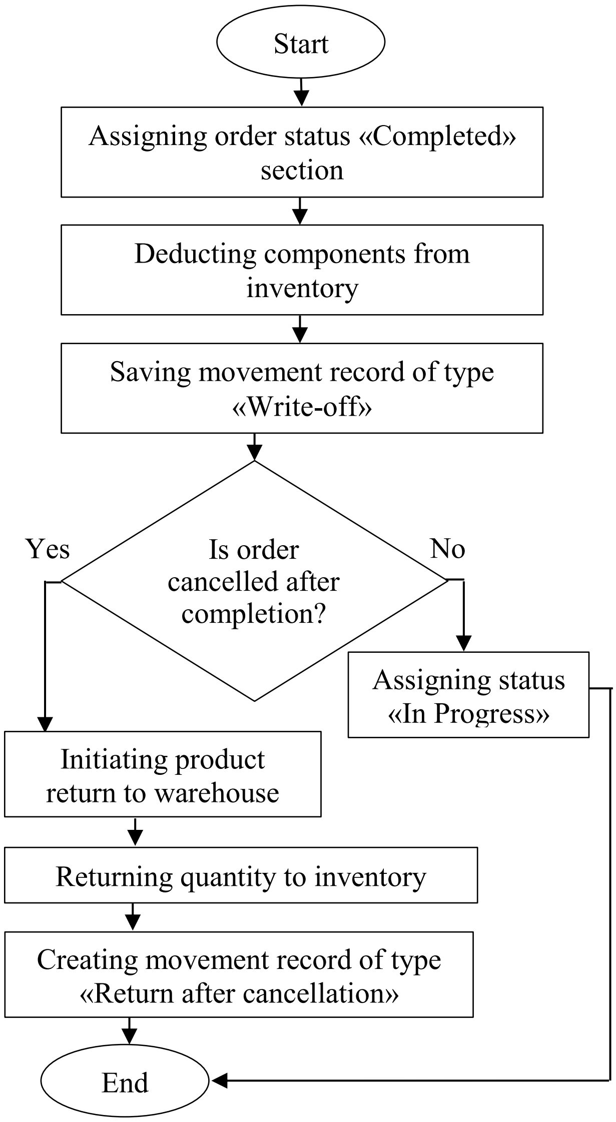


Fig. 9: Algorithm for updating balances when the order status changes

Source: created by the authors.

After the write-off procedure is completed, the system checks whether the order has been canceled after its execution. This can happen if the customer refuses an already prepared order or if other unforeseen circumstances arise.

If the order is not canceled, the algorithm completes, and all changes remain recorded in the system.

If the order is canceled after execution, the system initiates the process of returning the product to the warehouse. At this stage, a reverse process to write off is triggered.

The quantity of components that were previously written off is returned to the warehouse stock. The system adds this quantity to the current inventory, restoring the previous state of component availability.

A new movement record of the type «Return After Cancellation» is created to document this operation. This record documents the fact that the components have been returned to the warehouse, which allows maintaining a complete history of operations and provides the ability to audit all changes in warehouse inventory.

The algorithm is completed after the inventories are successfully updated and the corresponding component movement records are saved in the system database.

The developed algorithms form a complete functional cycle of the CRM system to support the engineering selection of components. The implemented architecture ensures the automation of key business processes: from user authentication and working with the component catalog to order processing, handling by managers, and automatic updating of inventory levels. The rule-based automated selection module allows processing both structured requests through forms and free-text requests, enhancing the system's flexibility. Integration of all components based on a three-tier architecture provides reliability, scalability, and the possibility of further system functionality expansion.

The logic of all the described algorithms and functional modules of the system is based on a well-designed data structure. Below is a description of the key entities of the relational database that implement the business model of the CRM system and ensure its operation.

The CRM system uses a PostgreSQL relational database with the following main entities:

- Users – stores user accounts with role-based access control (administrator, manager, client). Each user has credentials, contact information, and an assigned role that determines the available functionality;

- Components – central catalog containing sensors and actuators with technical specifications. Key attributes: item number, category (equipment, transport, sensors, actuators), subcategory, brand, technical parameters (temperature range, load capacity, interface type, IP protection class), and prices. Components are organized hierarchically through category-subcategory relationships;

- Projects – configurations created by clients that combine several components. Projects serve as templates for orders and store the quantity of components through an intermediate entity, ProjectComponent (many-to-many relationship);

- Orders – formalized requests based on projects. Each order tracks its status (new, in progress, completed, canceled), assigned manager, creation date, and the associated project structure. Changing the status automatically triggers the movement of components in the warehouse;

- Chats & Messages – a bidirectional communication system between clients and managers. Chat entities store the status (open/closed) and automatically open when clients send messages to closed conversations. Messages have timestamps and support read receipts;

- Inventory management consists of the entities Movement (individual warehouse operations) and MovementBatch (group operations). Each movement records the type (incoming/outgoing), quantity, date, and associated component. Stock levels are calculated dynamically by summing incoming and outgoing movements. The database schema ensures data integrity through foreign key constraints, supports complex filtering queries, and provides efficient reporting. The modular structure of the database allows for further expansion of system functionality without the need to modify its core architecture.

A visual representation of the described modular structure and the relationships between entities is an entity-relationship diagram (ER diagram). Below (Figure 10) is an ER diagram of the CRM system architecture, which visualizes the key data objects, their attributes, and the relationships that implement the functionality of engineering selection.

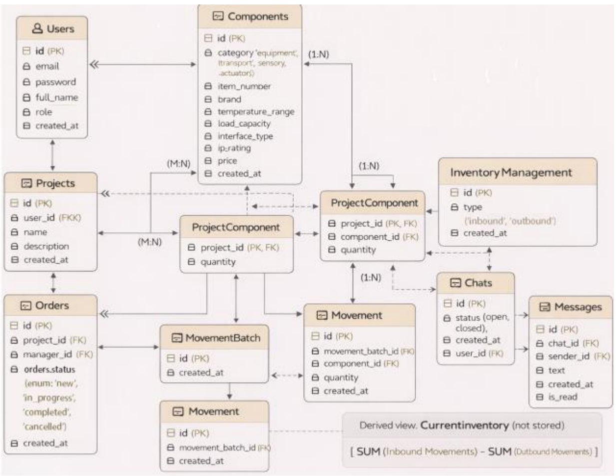


Fig. 10: ER diagram of the CRM system architecture for the engineering selection of technical components

Source: created by the authors.

Figure 10 shows the ER diagram of the CRM system, reflecting the modular relational database architecture. The diagram illustrates the key entities, their attributes, and relationships that provide the functionality for engineering component selection, project management, orders, inventory management, and user communication. The database structure follows the principles of normalization, ensures data integrity, and supports scalability without altering the underlying architecture.

4 Research Results and Discussion

The key outcome of the study is the successfully implemented three-tier architecture, which provided a clear separation of responsibilities and enhanced system maintainability. Developing the server-side using the Django framework allowed for the effective translation of formalized requirements into specific business models in Table 1.

Table 1. Development of the server-side based on the Django framework

Architectural solution	Technical implementation	Achieved effect
Flexible data model for components and configurations	- the *Component* model with fields *category* and *subcategory* (selection from *CHOICES*); - an intermediate *ProjectComponent* model for the many-to-many relationship between *Project* and *Component* with a *quantity*.	- provided the ability for multi-criteria filtering by category, subcategory, and technical parameters; - implemented storage of the project bill of materials (BOM) with a specific quantity of each component.
Automation of warehouse inventory updates	- overriding the *save()* method of the *Order* model; - automatic creation of records in the *Movement* model when the order status changes to «Completed» (write-off) or «Cancelled» (return).	- eliminated manual work and the risk of human error when adjusting balances; - guaranteed data integrity, meaning orders and stock status are always synchronized.

Effective calculation of current balances	- a static method *get_stock()* of the *Movement* model, which calculates the balance as the difference between the sum of receipts (*in*) and write-offs (*out*) for a specific component; - the *stock* property in the *Component* model (computed field).	- avoiding data duplication – no need for a separate field *current_quantity*, which requires constant synchronization; - full audit trail, meaning the balance is the result of all historical transactions, ensuring transparency; - performance because the aggregate queries to the database are optimized.

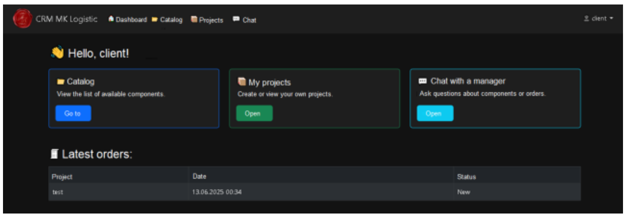
Source: created by the authors.

Thus, the presented architectural solutions demonstrate the transition from describing requirements to their software implementation. Each item in Table 1 addresses a specific challenge: the data model formalizes knowledge, automation eliminates routine tasks, and inventory calculations build trust in the system through transparency. This is not just a technical implementation – it is the creation of a digital foundation for engineering solutions.

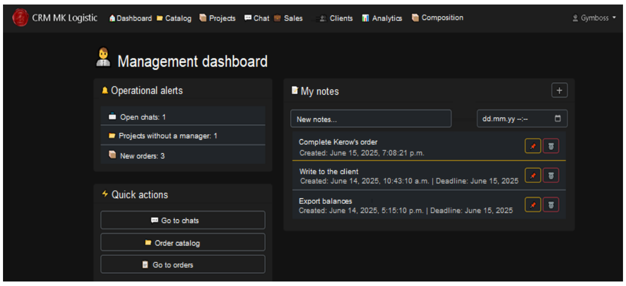
Choosing Django and PostgreSQL proved to be optimal for rapid prototyping with complex logic. The *Movement/MovementBatch* model provided full audit traceability of changes, which is critical for logistics systems. A future step could be moving part of the logic into separate services (microservices architecture) for even better scalability.

The next critical stage of the research was the implementation of a safe and ergonomic interface for different categories of users. While the server architecture is responsible for the integrity of business logic and data, the access control system and interface design determine the practical usability of the system for end users – engineers, managers, and administrators. Implementing a role-based access model using a custom Django user model with a role field allowed for a clear and secure management system to be created. Empirical testing confirmed its effectiveness:

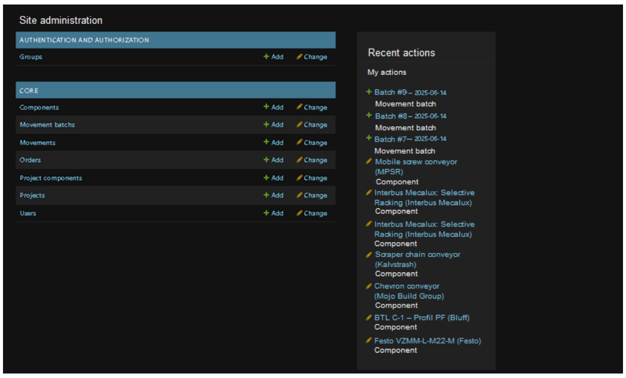
- for the client, the interface (Figure 11, a) is focused on self-service tasks: catalog search, project formation, communication via chat. This helps reduce the workload on managers at early stages;
- for the manager, the dashboard (Figure 11, b) aggregates key performance indicators. Centralized interfaces for working with chats, orders, and analytics ensure prompt task resolution. The «Quick Actions» and «Notes» modules enhance personal efficiency;
- for the administrator, the panel (Figure 11, c) provides full control over data, users, and inventory, serving a «back-office» function.



a)



b)



c)

Fig. 11: CRM system user interfaces for different roles: a) client; b) manager; c) administrator

Source: created by the authors.

A more detailed description of the client interface, manager dashboard, and administrator panel will be provided in our subsequent studies.

Such differentiation has led to the creation of intuitive and ergonomic interfaces, adapted to specific usage scenarios. This meets the non-functional requirement NFR1 (Usability). The absence of redundant information in each interface reduces the user's cognitive load and shortens the learning time. The proposed model can be extended by adding more detailed permissions within roles.

To verify the correctness of the implemented logic, 24 automated test scenarios were created and executed (Figure 12). The testing covered critical aspects of the system, including order and inventory logic (8 tests), chat functionality (6 tests), access control (5 tests), and data integrity (5 tests). The tests confirmed the correctness of the automatic creation of *Movements* when the order status changes. The correctness of automatically reopening a closed chat upon receiving a new message from a client was also verified, which is implemented in the algorithm of the *Message.save()* method. It was separately checked that users with different roles have access only to the views assigned to them. In addition, the correctness of relationships between key models was tested, for example, the process of adding components to a project through the intermediate *ProjectComponent* model.

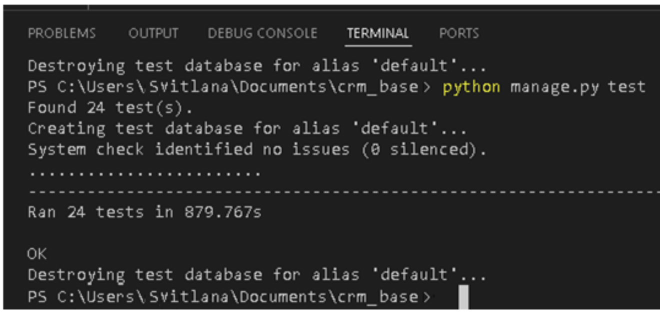


Fig. 12: CRM System Testing Results

Source: created by the authors.

The results of all tests showed 100 % successful execution in 0,665 seconds (Figure 12), empirically confirming the stability, compliance with requirements, and technical reliability of the system core.

Having established the technical correctness of the system, it is reasonable to assess its practical value compared to existing market solutions. Comparison with general-purpose CRMs (KeyCRM, Zoho CRM, SuiteCRM) based on specialized criteria (Table 2) revealed the competitive advantages of the developed solution.

Table 2. Comparison Matrix of CRM System Functionalities

№	Criterion	Proposed CRM	Key CRM	Zoho CRM	W
A. Basic CRM functionality					
1	Customer management	+	+	+	0,077
2	Role-based access	+	+	+	0,077
3	Communication with clients	+	+	+	0,062
4	Analytics and reporting	+	+	+	0,062
B. Engineering functionality					
5	Catalog of technical components	+	+	+/-	0,115
6	Automated component selection	+	-	-	0,154
7	Compatibility check	+	-	-	0,115
8	BOM Management	+	+/-	+/-	0,092
C. Technical capabilities					
9	Warehouse management	+	+	+/-	0,077
10	Data Import (YAML/CSV)	+	+/-	+/-	0,053
11	API та інтеграції	-	+	+	0,062
12	Customization	+	+/-	+/-	0,054
Considered assessment		0,938	0,632	0,536	

Source: created by the authors.

Note. W – the weight of the criteria, determined by an expert method taking into account the specifics of engineering component selection. Criteria 6-8 received the highest weights (0,20 and 0,15) as they are unique for the target application.

The rating «+/-» equals 0,5 points and is assigned in cases when:

- the functionality requires additional modules or settings;
- the implementation is incomplete or limited compared to a full solution;
- there is basic support, but without specialization for technical tasks.

The rating «+» equals 1, and the rating «-» equals 0.

Thus, the overall score is calculated by multiplying the numerical value of the rating by the weighting coefficient and then summing all these values. The comparison conducted using the weighted rating method (Table 2) clearly demonstrates that the proposed specialized CRM system significantly outperforms the general platforms KeyCRM and Zoho CRM in terms of overall effectiveness for engineering component selection tasks, achieving the maximum weighted score of 0,938.

To confirm the improvement in the efficiency of engineering selection, an experimental validation was conducted with 5 users (design engineers). The methodology was as follows. Each participant completed 3 typical component selection tasks: using the traditional method (Excel spreadsheets + PDF catalogs); using the developed CRM system.

Thus, each design engineer completed 3 typical component selection tasks for real projects:

1. Selection of a temperature sensor for a refrigerated chamber with an operating range of -30 °C to +5 °C.
2. Choice of an actuator for a conveyor line with a maximum load of up to 50 kg.
3. Configuration of a warehouse monitoring system, which includes five sensors and a control controller. The results are provided in Table 3.

Table 3. Comparative evaluation of component selection efficiency

Metric	Traditional method	Proposed CRM	Result
Average selection time	38 min	15 min	↓ 61 %
Number of compatibility errors	18 %	5 %	↓ 72 %
BOM Formation Time	12 min	3 min	↓ 75 %

Source: created by the authors.

The improvement values represent the relative reduction of time and errors achieved by the developed CRM system compared to the traditional selection approach.

The developed CRM system ensures a 61 % reduction in engineering selection time and a 72 % decrease in the number of errors, confirming the achievement of the stated goal of increasing efficiency.

The system architecture is designed with scalability and extensibility principles in mind. The three-tier model with a clear separation of presentation, business logic, and data layers enables future integration with external CAD/PLM/ERP systems via a RESTful API. The current version focuses on implementing the critical core functionality (FR1-FR3): specialized engineering component selection, compatibility verification, and component ranking. This aligns with an iterative development approach, where the core decision support capabilities are realized in the first version, creating a foundation for subsequent integrations.

Thus, the developed CRM system not only automates standard business processes but also acts as an engineering decision support tool. The formalization of component characteristics, compatibility rules, and ranking criteria reduces subjectivity in engineering selection, improves repeatability of decisions, and decreases the time required for component configuration. This directly fulfills the stated goal of increasing the efficiency and reliability of engineering component selection for automated systems.

5 Conclusion

This study was dedicated to the development and investigation of a specialized CRM system designed to support the engineering selection of technical components of automated systems. Unlike traditional CRM solutions, which are primarily focused on marketing and commercial processes, the proposed system is viewed as an information platform for managing engineering data, users, and decision-support processes related to the selection of technical components.

During the course of this work, the following main results were obtained.

Formalization of the engineering component selection problem. In Section 2, a formal formulation of the technical component selection problem is proposed in the form of a multi-criteria analysis and optimization problem. Sets of components C, types K, characteristics H, and requirements R are defined. The process of basic selection (formulas 1-5), compatibility checking through the BCR relation (formulas 6-8), and ranking based on the fitness function $F(c_i, R)$ (formulas 9-11) are formalized. The resulting model provides a mathematical foundation for algorithmic implementation of automated engineering selection within an information system.

Formalization of requirements for the CRM system. Section 3.1 defines the functional (FR1-FR4) and non-functional (NFR1-NFR3) requirements for the system in accordance with the ISO/IEC/IEEE 29148:2018 standard. Functional requirements include: multi-criteria search and filtering (FR1), project management and selection history tracking (FR2), automated compatibility checking (FR3), and component ranking (FR4). Non-functional requirements include usability (NFR1), performance < 3 seconds (NFR2), and scalability (NFR3).

Development of the architecture of a CRM system for engineering selection support. In section 3.2, a three-tier CRM system architecture is developed, which combines the classic functionality of managing users, projects, and orders with specialized engineering subsystems for automatic component selection, compatibility checking, and results ranking. This approach allows the integration of decision support algorithms into the standard CRM environment without disrupting its basic architecture.

Software implementation of the system. Section 3.3 describes the implementation of a CRM system based on the Django framework and the PostgreSQL relational database. A set of algorithms has been implemented, covering the full lifecycle of engineering selection: from requirements formation and component selection to compatibility verification, project creation, specification (BOM) generation, and automatic inventory operations management. The chosen technological platform ensures scalability, modularity, and the possibility of further functionality expansion.

Chapter 4 presents an experimental evaluation of the effectiveness of the developed CRM system involving design engineers performing typical component selection tasks. A comparative analysis with the traditional approach (using spreadsheets and PDF catalogs) showed a significant improvement in engineering selection efficiency. In particular, the average time for component selection was reduced by 61 %, the number of compatibility errors decreased by 72 %, and the time required to generate a bill of materials (BOM) was reduced by 75 %.

The conducted comparative analysis demonstrated that the developed CRM system outperforms general-purpose CRM platforms due to the presence of specialized engineering functionality aimed at automated selection, compatibility analysis, and support for technical decision-making.

The novelty can be considered as:

1. Formalization of the problem of engineering component selection as a multi-criteria optimization problem with decision support (formulas 1–13), which is adapted to a CRM architecture for the first time.

2. Developing a hybrid approach to automated component selection, combining a rule-based system with the processing of structured and text user queries and automated compatibility checking according to the BCR relationship.

3. Development of a specialized CRM architecture for engineering tasks that integrates modules for multi-criteria analysis, compatibility checking, and ranking into a single data management information system.

4. Providing a quantitative assessment of the efficiency of the selection and ranking algorithms. The scientific contribution of the work lies in formalizing the engineering component selection problem as a multi-criteria optimization task with compatibility constraints, which is modeled as a CSP. To solve this problem, a detailed ranking algorithm based on a weighted sum with an analyzed computational complexity of $O(N \log N)$ is proposed, ensuring its practical effectiveness for working with large-scale catalogs containing tens of thousands of components.

Unlike existing CRM solutions focused on business processes, the proposed approach introduces an engineering-oriented decision support model with clearly defined computational complexity and experimentally validated effectiveness.

The practical significance of the results obtained lies in the possibility of implementing the developed CRM system in engineering and design organizations to reduce design time, decrease the number of errors, and increase engineers' productivity.

Directions for further research include integrating the system with CAD/PLM platforms via a RESTful API, expanding the automatic selection subsystem using machine learning methods, developing an ontological model for the compatibility of technical components, and scaling the system to the enterprise level.

Thus, the research goal has been achieved: the developed CRM system with a decision support module provides increased efficiency in the engineering selection of technical components for automated systems.

References:

[1] B. Kocaoglu, Digital transformation in logistics. *In: Logistics Information Systems: Digital Transformation and Supply Chain Applications in the 4.0 Era. Cham: Springer Nature Switzerland*, 2024, pp. 1–35

[2] S. Ghosh, M. Hughes, I. Hodgkinson, P. Hughes, Digital transformation of industrial businesses: A dynamic capability approach, *Technovation*, 2022, 113, pp. 102414. DOI:10.1016/j.technovation.2021.102414

[3] M. M. Mikhlif, S. Smaoui, The impact of digital transformation on accounting information systems. *Asian Journal of Management and Commerce*, 2024, 5(1), pp. 289–294. DOI:10.22271/27084515.2024.v5.i1d.272

[4] C. Lin, J. Chen, C. Feng, X. Li, Optimizing supplier selection for prefabricated components: a comprehensive evaluation, *Engineering, Construction and Architectural Management*, 2026, 33.2, pp. 1649–1677. DOI:10.1108/ECAM-09-2024-1230

[5] M. Rudenko, S. Sotnik, Classification of CRM systems, *Manufacturing & Mechatronic Systems 2025: Proceedings of IX st International Conference, Kharkiv, October 25-26, 2025: Thesises of Reports*, 2025, pp. 42–45

[6] M. Rudenko, S. Sotnik, Classification of CRM systems, *Manufacturing & Mechatronic Systems 2025: Proceedings of IX st International Conference, Kharkiv, October 25-26, 2025: Thesises of Report*, 2025, pp. 42–45

[7] T. Stefanov, S. Varbanova, M. Stefanova, I. Ivanov, CRM System as a necessary tool for managing commercial and production processes, *TEM Journal*, 2023, 12(2), pp. 785–797. DOI: 10.18421/TEM122-23

[8] N. H. Tien, D. T. P. Chi, N. M. Duc, T. T. K. Hong, The formation and development of CRM system at Thien Hoa electronics supermarket in Vietnam, *International Journal of Research and Growth Evaluation*, 2021, 2(4), 752–760

[9] P. Agarwal, A. Gupta, Harnessing the power of enterprise resource planning (ERP) and customer relationship management (CRM) systems for sustainable business practices, *International Journal of Computer Trends and Technology*, 2024, 72(4), pp. 102–110. DOI:10.14445/22312803/IJCTT-V72I4P113

[10] O. Berestetska, T. Iankovets, A. Orozonova, S. Voitovych, A. Parmanasova Using CRM systems for the development and implementation of communication strategies for digital brand management and internet marketing: EU experience, *International Journal of Professional Business Review: Int. J. Prof. Bus. Rev.*, 2023, *8*(4), pp. 1–16. DOI:10.26668/businessreview/2023.v8i4.1613

[11] O. Hamza, A. Collins, A. Eweje, G. O. Babatunde, Agile-DevOps synergy for Salesforce CRM deployment: Bridging customer relationship management with network automation, *International Journal of Multidisciplinary Research and Growth Evaluation*, 2023, 4(1), pp. 668–681. DOI:10.54660/.IJMRGE.2023.4.1.668-681

[12] B. Khashab, S. Gulliver, R. Ayoubi, C. Strong, Analysing enterprise resources for developing CRM framework in higher education institutions, *Journal of Enterprise Information Management*, 2022, 35(6), 1639–1657. DOI:10.1108/JEIM-09-2021-0421

[13] S. Koli, R. Singh, P. Rana, A. Aggarwal, A. Dumka, Salesforce technology: A complete CRM solution on the cloud, *2023 2nd Edition of IEEE Delhi Section Flagship Conference (DELCON)*, 2023, pp. 1–5. DOI:10.1109/DELCON57910.2023.10127497

[14] E. Gołąb-Andrzejak AI-powered Customer Relationship Management–GenerativeAI-based CRM–Einstein GPT, Sugar CRM, and MS Dynamics 365, *Procedia Computer Science*, 2024, 246, 1790–1799. DOI:10.1016/j.procs.2024.09.683

[15] N. Huttunen, Exploring how businesses benefit from CRM systems: A comparative study measuring the performance of Salesforce, HubSpot and Microsoft Dynamics 365, *Examensarbete Företagsekonomi*, 2024, 51 p. [Online]. Available: https://urn.fi/URN:NBN:fi:amk-2024121837315 [Accessed: Mar. 2, 2026]

[16] O. Iwalehin, D. M. Fisher, CRM With Salesforce HubSpot and Motion AI: Chatbot-enabled CRM–An Analysis of Automated Customer Relationship Managemen, *Iconic Research and Engineering Journals*, 2023, 7(5), pp. 301–303. [Online]. Available: https://www.irejournals.com/paper-details/1706524 [Accessed: Mar. 2, 2026]

[17] M. Chaikovska, Modern convergent challenges in IT project management for CRM system implementation in Ukraine, *Știință, educație, cultură*, 2025, Vol. 1, pp. 157–163. DOI:10.5281/zenodo.15176683

[18] A. Puška, Selection of CRM systems using objective criteria for the needs of small companies, *International Journal of Knowledge and Innovation Studies*, 2025, 3.1, pp. 1–11. DOI: 10.56578/ijkis030101

[19] U. Shylovich, Particularities of CRM-systems implementation in small business, http://edoc.bseu.by, 2022, pp. 1–2

[20] M. Pohludca, H. Stvercova, The Best Practice of CRM Implementation for Small- and Medium-Sized Enterprises, *Administrative Sciences*, 2019, № 12, pp. 22–39. DOI: 10.3390/admsci9010022
[21] S. Harutyunyan Gor, TTechnological innovations and systems in the field of customer relationship management (CRM), *Rehyon*, 2025, 2, pp. 57. DOI:10.58587/18292437-2025.2-225
[22] P. N. Dattatraya, B. Chandra, Scalability Challenges and Solutions in CRM Automation for Small-Scale Enterprises: A Comparative Study, *Journal of Integrated Engineering Sciences*, 2025, 1(4), pp. 45–59. DOI:10.63686/jies.v1i4019
[23] N. Hagenauer, Instructions on how to work with Siemens Teamcenter for university purpose, *Thesis Mechanical Engineering Bachelor*, 2023, 84 p. [Online]. Available: https://urn.fi/URN:NBN:fi:amk-2023052313335 [Accessed: Mar. 2, 2026]
[24] S. Dertien, W. Hastings, The state of digital thread, *PTC White Paper*, 2021, 1–16. [Online]. Available: https://www.cimdata.com/images/PLMRoadMap/The-State-of-Digital-Thread-0721.pdf [Accessed: Mar. 2, 2026].
[25] A. Prasad, Transforming Product Development with a Platform-Based Approach to Product Lifecycle Management, *SAE Technical Paper,* 2025, pp. 1–13, DOI:10.4271/2025-01-5051

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)
The authors equally contributed in present research, at all stages from the formulation of problem to final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself
Self-financing.

Conflict of Interest
No conflicts of interest in this research.