

# Intelligent System for Socialization of Individual's with Shared Interests based on NLP, Machine Learning and SEO Technologies

Taras Batiuk<sup>1</sup>, Victoria Vysotska<sup>1,2</sup>, Roman Holoshchuk<sup>1</sup>, Svitlana Holoshchuk<sup>1</sup>

<sup>1</sup> Lviv Polytechnic National University, S. Bandera Street, 12, Lviv, 79013, Ukraine

<sup>2</sup> Osnabrück University, Friedrich-Janssen-Str. 1, Osnabrück, 49076, Germany

## Abstract

The main objective of this work is to create an intelligent system for socialization by personal interests based on SEO technologies and methods of machine learning. The primary purpose of this intelligent system is to identify the user within the system using neural networks and to select similar users by analyzing the user's current information. An intelligent system is created that, through Identity and JWT tokens, provides optimized and secure authorization, logging, and support functions for the current system user session. Finding a face in a user's photo and checking the presence of a similar user in the database are implemented using convolutional and Siamese neural networks. The analysis and formation of similar user beeps are implemented using fuzzy search algorithms, the Levenshtein algorithm, and the Noisy Channel model, which made it possible to maximize the user selection process's automation and optimize the time spent in this process. Tools have also been created to view other user's profiles, preferences and private correspondence. All personal mail and information about it are stored in the current database. Each user of the system can view all the information about sent and received messages. The created intelligent system meets the original goal, as it fully implements user identification, analysis, selection and further socialization of users of the system. The algorithm of sampling of users of similar interests implemented in the IS is 25-30% more efficient and accurate than usual Levenshtein algorithm. At the same time, the implemented algorithm performs sampling 10 times faster than Levenshtein algorithm.

## Keywords 1

Levenshtein distance, Siamese neural network, convolutional neural network, fuzzy search, Noisy channel

## 1. Introduction

Developing an intelligent system for the socialization of individuals is one of the critical tasks of modern social communication. Individuals attempt to optimize all life processes to save time and use it in the most efficient and optimized way. When searching for specific programs, users first choose those that save time, are simple and easy to use, optimize their work, and are automated enough to perform most actions instead of the system user. First of all, it is worth noting the time savings. Every day, people use at least dozens or even hundreds of different software systems, including smartphone applications, computer programs for the desktop, browser-based web applications, and many other programs. Almost every program has several analogues, giving a wide choice for the user and choosing the most convenient program. Understanding complex programs requiring a detailed acquaintance with all the system functions do not appeal to any potential users.

The main goal of our research is to create an intelligent system for the socialization of individuals that uses SEO technologies and machine learning methods. SEO technologies include the fuzzy word

---

COLINS-2022: 6th International Conference on Computational Linguistics and Intelligent Systems, May 12–13, 2022, Gliwice, Poland  
EMAIL: thatjuk4u@gmail.com (T. Batiuk); victoria.a.vysotska@lpnu.ua (V. Vysotska); roman.o.holoshchuk@lpnu.ua (R. Holoshchuk);  
svitlana.l.holoshchuk@lpnu.ua (S. Holoshchuk)  
ORCID: 0000-0001-5797-594X (T. Batiuk); 0000-0001-6417-3689 (V. Vysotska); 00-0002-1811-3025 (R. Holoshchuk); 0000-0001-9621-9688 (S. Holoshchuk)



© 2022 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).  
CEUR Workshop Proceedings (CEUR-WS.org)

search algorithms necessary for the system to function using the Noisy Channel model with algorithms for efficient distribution of text information. Machine learning methods in this situation include a convolutional neural network for identifying system users. Accordingly, a system that uses modern technologies to perform its primary functions will be created. Such an intelligent system is available to almost all users, as it will be implemented as a browser-based web application. The system will allow all its users to register quickly, enter basic information necessary for the system to work and start convenient socialization with other system users.

The research object is the socialization of individuals, which is very important. All modern social networks try to optimize and automate the socialization of various users as much as possible using all popular modern technologies, such as neural networks and algorithms for analyzing user text. For the successful creation of an intelligent system for the socialization of individuals by common interests, the most critical task is to correctly understand and work out the process of socialization of users.

The subject of the research, in this case, is located in the middle of the object of research; it is the user of the intelligent system, whose primary goal is to carry out socialization. Accordingly, the system user is studied, namely, determining the user's authenticity by searching for a human face in a user's photos with the help of neural networks, fuzzy search algorithms, and the Noisy Channel model.

According to the source text, the scientific novelty of the obtained results includes designing a new algorithm to analyze user's information and search for the most suitable users. It is based on existing algorithms such as the Levenshtein algorithm, the sampling algorithm for data collection, the N-Gram algorithm, and the Noisy Channel model. The asynchronous programming template is developed, which makes it possible to create a completely dynamic system. The convolutional neural network, which facilitates an adequate search for human faces in photos and checks the people in the system's database, has been improved. The practical significance of the developed intelligent approach for the socialization of individuals is fundamental. First of all, it is socially meaningful since it performs a vital function of socialization of individuals, which is very necessary for our time. It also allows all users of the system to quickly and conveniently use all its parts.

Moreover, it is an innovative system as there are no similar analogues for the socialization of individuals who would use such algorithms and have an equivalent automation level. The system selects, analyzes, and processes text data and generates the final result efficiently and quickly. The system uses SEO technologies as any system performs the functions of searching and processing data. The neural network allows identifying the user based on their photo effectively. In general, developed algorithms enable the creation of a convenient intelligent system using the necessary algorithms. Optimizing the smart strategy is a complete asynchrony that avoids long waits and complicated queries in processing and analyzing. It makes the system work with different amounts of data, analyze them, process and generate new ones effectively and dynamically. Besides that, a cloud service that distributes data is applied. It will store all the heaviest data in the cloud environment and use a simple software interface to download all the necessary data.

Thus, we may argue that the development of the intelligent system is essential both in social terms and the implementation of all algorithms that provide its necessary functionality.

## **2. Related works**

Machine learning and SEO technologies are currently significant in the process of developing intelligent systems. Almost every software system used by a large number of people possesses specific socialization mechanisms. For the correct and effective implementation of these mechanisms, it is necessary to optimize existing algorithms and design new ones to meet rigid specifications. A significant number of works study the optimization and implementation of machine learning algorithms, socialization, and SEO technologies. Some of them deal with the analysis of user's interaction within the social networks [1]. In the modern world, most people use social networks and spend a considerable part of their free time there, so analyzing their behaviour in certain situations while using social networks is essential and relevant. We study the main aspects of user interaction with each other. It involves analyzing sent messages and replies to them, which allows specifying certain system users' patterns and parameters and further automate the system operation. The implementation of mechanisms for analyzing and checking the current online user experience is highlighted. It is a significant point for

dynamic social networks that track the main user actions to optimize the operation of the system mechanisms. Fundamental issues of moderation of internet content within social networks are considered to effectively analyze user texts and respond to specific violations of the tools.

In the article [2], the authors highlight a sample of data from analyzing social networks among different user groups by using paired parameters. The main results are received from the user interaction and studied based on their message communication. The use of a particular framework for the socialization of users in the social network *Twitter* is considered. The main features of hashtag formation and generated hashtags by users with different input parameters are studied.

We elaborated on the main questions that should be answered and algorithms that should be optimized to analyze possible information about the users more effectively. The concepts of additional algorithms that affect user information analysis are also developed in case of implementation.

The authors reflect important software issues in the article [3]; namely, the principles of building and maintaining efficient server systems that allow to withstand a large number of ongoing users and keep the system in a stable state, regardless of external factors, are considered. This problem is highly relevant since many people use social networks from phones and computers, and the number of users of the popular social networks can reach millions at a time. Another highly relevant issue for our work is to ensure the processing and information output for all users regardless of which systems and devices they have or what the internet speed is. The basic principles of building highly efficient server systems that allow parallel requests and, accordingly, can process requests asynchronously, forming a response to a request in particles, dynamically storing intermediate information are described. Ways to save the information and determine its reliability left by the user are considered; the main ways to avoid loss and distortion of information are highlighted, and ways to optimize the operation of the data storage system are taken into account. The main concepts of using neural networks in user interaction within social networks are studied, and the principal and effective forms of user text analysis are found out.

The primary information obtained from the neural network analysis of user messages left on *Instagram* is highlighted. We also analyzed the length of written letters and the factors which affect them. In the article [4], the author considers that specific mechanism designed for interaction between client and server systems highlight the main features of transmitting and storing intermediate information as efficiently as possible, avoiding data loss and forming middle requests from users. Despite the many tasks that solved ineffective social networks, many functions are developed to solve the problems. The main is multithreaded user data processing without losing system efficiency and asynchronous processing of incoming requests. They allow separating the client part from the server and transfer some work to the client to optimize the system. There is also an important task to fully implement the cross-domain data transmission model, which will maximize the formation of a request to the server and respond to the request due to generating and transmitting the current request.

In the article [5], the authors discuss the use of neural networks in social networks. Many people use social networks in the modern world, watch videos, listen to audio, upload and view photos. Neural networks are often used to automate and optimize specific actions. They have a wide range of applications, analysis of audio, video, photo formation, and analysis of text information. All this is hugely relevant in our time and actively studied in the process of creating software systems. Implementing a neural network that can set the basic needs of users by analyzing the list of their online purchases using specific parameters is considered. It shows how an implementation of a neural network can help identify users' moods and emotions out of messages left by them on social networks.

In the article [6], the author highlights how to determine the presence of users in a photo in the social network *Instagram* and proposes mechanisms for optimizing photo analysis using neural networks. The basic principles of neural networks for identifying various interconnected objects in selected photos on the examples of images with food are considered. The main mechanisms of interaction between users in social networks are defined, and a neural network that forms a sample of messages using the specified parameters is proposed. Neural networks are widely used; despite the demonstrated capabilities in analyzing text information, audio data, and photos, many problems need to be solved. The best solution is to use neural networks, which is a rather challenging task. It involves developing a mathematical model of the system from scratch and implementing its proper and practical testing. It will optimize the work of social networks and use convenient user interaction functions.

Interaction and analysis of users, their requests, and entered information in various socialization systems and online stores are essential issues that the author considers in the article [7]. Nowadays,

there is a crucial task for online stores – they should optimize the product offer they want to find. To do this, one needs to analyze users. Their search queries create a particular sample of users for analysis and use algorithms to select the necessary information from a certain amount of text data. The article shows mechanisms for finding users based on specific normalized parameters and the subsequent formatting of the obtained data to automate user interaction. The basic principles of processing media files uploaded by users, namely photos and videos, are considered, and the basic principles of asynchronous storage of user data are described. The main algorithms of linear search of user information for determining the main parameters of interaction of people and factors that affect the data in user profiles are highlighted with the further data analysis. Implementing a neural network that determines users' forecasts and the results of football matches by analyzing pages in social networks of football clubs and user comments are considered.

The mechanism of adequate socialization of users is developed using the functionality of setting user preferences and optimizing the main processes of user interaction as a result of mutual socialization in one of the social networks. Implementing the primary mechanisms for determining and analyzing user needs using convolutional neural networks and algorithms for analyzing text data and determining and comparing the main selected parameters for a model user is highlighted.

In the article [8], the authors consider the main mechanisms of socialization of users who use the same hashtags in the social network *Twitter* and identify the features of their interaction.

The system features that analyze users' social networks and use fuzzy search algorithms among text data to determine the primary user needs are shown. Despite some solved problems, the problem of analyzing and processing user tools is still urgent. As a result of data separation and extraction, text analysis algorithms are constantly optimized and created using new algorithms.

Analysis and processing of user text information in various software systems and further determining user emotions and prediction of specific actions were brightly highlighted by the authors in the article [9]. Implementing a neural network that determines the emotions and mood of users by analyzing several photos of the user, normalizing information, and forming initial data is considered.

The main mechanisms of user socialization in the social network *Facebook* and its implementation are highlighted, and algorithms for user interaction are optimized over a certain period.

The features of analyzing user reviews for purchased products, implementing a sampling mechanism, and determining a certain percentage of irrelevant reviews are shown.

The main algorithms for analyzing the user's social networks that determine the user's primary emotions over a particular time are considered, and the fuzzy search algorithm is optimized. A mechanism for displaying targeted ads to the system user has been developed following the user's basic information, which is analyzed using fuzzy search algorithms. A selection of parameters for each user has been generated. Algorithms for analyzing server data and their further storage in the database are highlighted, and algorithms for storing data in the database in case of system load are developed.

In the article [10], the authors develop an algorithm for analyzing social networks of teenagers, design a neural network that determines the main parameters of teenagers, and optimizes their communication. Although the authors solve several problems, text analysis and determining the user's emotions are open. For information analysis, both neural networks searching for specific data in the text according to specified parameters and linear information search algorithms designed to process a large amount of text and analyze it being partially damaged are used.

Data analysis and further efficiency of information processing are very important for creating an intelligent system and are considered by the authors in the article [11]. In the modern world, all software systems work with different data. The result of processing is a set of extra data analyzed using specific algorithms or using neural networks. Processing, storage, and use of the received data is a crucial stage in its development. It is necessary to combine constant and dynamic data to optimize the operation of the software product and increase the efficiency of work. A mechanism for determining the primary user's outcome using a mobile device is developed using a convolutional neural network that analyzes the user's work with a mobile device and generates basic information about a given user sample.

The author [12] considers a neural network that allows determining the approximate age of a user of the social network *Instagram* by searching for the user's face in photos and processing them. A mechanism for analyzing media data, namely movies and TV shows, is invented using a neural network to determine the minimum age of viewers allowed for watching media content. The algorithm for low-quality media files is optimized. The technology of analyzing voice messages of social network users

is highlighted, and a neural network is developed that determines the user's emotions by analyzing these voice messages. The mechanism of saving and processing the user history of product searches in the online store is considered. A means for offering the most relevant products to the user is developed.

A mechanism for user socialization in the social network *Facebook* using the Levenshtein algorithm and a tool of asynchronous data storage has been developed. An optimized algorithm for fuzzy search of certain information when performing data search and text analysis is highlighted, and an optimized mechanism for asynchronous text analysis is developed. Despite the problem's solution, it is still vital to optimize the operation of system components and create new algorithms for further processing of the obtained data, making software systems more flexible and accessible.

The authors consider the use of information search algorithms in text data in the article [13]. It is essential to use the Levenshtein and fuzzy search algorithms. These algorithms are also flexible, enabling creating a separate implementation of the algorithm following the problem. The performance of modern information search mechanisms in social networks is considered, innovative algorithms that require optimization for faster data search and processing are presented, and algorithms for optimized data storage are highlighted. Mechanisms for analyzing user requests in online stores are studied. An algorithm for generating data generated as a result of query analysis is given. The ways of using the generated data to save information about a specific user are created. The implementation of algorithms for asynchronous dynamic calculation and data processing is specified. A method for partial data storage that avoids repeated overloading of the Web page and creates a more dynamic intelligent system is developed. The implementation of user socialization mechanisms using the information immobilization model is considered. It optimizes the data analysis process and performs analysis in several streams. The Levenshtein algorithm for user socialization is also optimized.

In the article [14], the author shows the mechanisms for analyzing user queries in online stores and considers implementing the fuzzy search algorithm among distorted data. It helps to restore the original lost information, select data parameters, and perform text analysis. The article also highlights the implementation of binary algorithms for data search in social networks. It forms the main aspects necessary to optimize data search and ensure the integrity of transmitted data under System load to protect them. As a result, we can say that there is still a problem of optimizing text data processing algorithms since each software system is unique. For each method, it is necessary to create a particular algorithm and optimize it according to the system's features.

The author highlighted the importance of using machine learning in modern Intelligent systems in the article [15]. Nowadays, all systems operate with a large amount of data; it is often necessary to distribute data to optimize and speed up system processes. It is achieved by processing part of the data using machine learning. Features of creating server programs are highlighted, algorithms for the interaction of server programs belonging to different hosts are optimized, an algorithm for optimized packet data transmission in particular objects is created, and algorithms for storing received objects are created. The implementation of text data analysis algorithms in social networks is considered. A neural network is created that compares user photos and text information left by the user and forms a sample based on the analyzed data, determines the type and main parameters of the system user, and optimizes the operation of search algorithms.

In the article [16], the author highlights the main algorithms for clustering data stored on mobile devices, forms the main patterns for saving processing, and developing a new data sample using the current one. It has optimized information search algorithms to avoid irrelevant data and create a helpful piece are presented.

The authors show the use of SEO technologies in software systems that involve user interaction in the article [17]. Nowadays, SEO technologies are essential for optimizing search algorithms and user socialization. An adequately implemented algorithm makes it possible to effectively use all the intelligent system features for searching and interacting with users to optimize the system's main processes. Implementing a linear algorithm for analyzing user text data is considered. An algorithm for constructing information diagrams created using the specified parameters for data processing and fuzzy search algorithms are optimized for a more accurate search of relevant information in a given text. The article highlights the implementation of data extraction mechanisms, creates a neural network that analyzes all social communication initially, and uses input parameters to throw out all irrelevant details saving only the necessary data sample at a particular moment. The implementation of data storage mechanisms from social networks in mobile devices is shown. Data storage on system servers and the

device's local database is optimized, and data transmission and efficient storage algorithms are worked out. The authors also propose several new algorithms for analyzing users' social networks, such as neural networks. They analyze photos, and audio recordings of users and algorithms for analyzing user messages that are extremely important in creating global web applications as automating user information processing is one of the most critical problems [18-21].

To sum up, we can say that using SEO technologies is essential since there are many algorithms characterized by the authors in articles [22-26], which need to be optimized for the effective use of intelligent systems of user socialization. We need to create new practical algorithms to improve the operation of modern intelligent systems and carry out effective data processing [27-32]. When creating the selected method, an important task is to study and analyze existing analogues. With the help of this research, you can identify the main advantages and disadvantages of existing intelligent systems and, when creating a design, avoid existing penalties and repeat the main benefits [33-37].

It is worth noting that almost no intelligent systems similar to the intelligent design of socialization of individuals are found during the search for analogues. Among the programs studied, it is worth highlighting three analogues that are most similar in their functionality to the created intelligent system. These are *Tinder*, *Badoo*, and *Chatous* programs. If we consider each of these programs separately, we can highlight the main advantages of using this program and its disadvantages [38-41].

*Tinder* of all three listed above is the most popular system for socializing individuals, and it is one of the oldest compared to other programs. Its main advantage is the presence on all platforms; *Tinder* is used as a desktop program on all operating systems, an application for a smartphone, and the browser version. Another advantage is the photo analysis, the search for analogues of the face in the *Tinder* system itself, and the face presence in the photo. It guarantees that the program is used only by real people. When you meet or correspond with the user, you are sure that it is not a scammer. The complete absence of advertising, which allows you not to be distracted, adds to the program's benefits.

Among the disadvantages, it is worth highlighting that the program uses only basic user search filters, namely age, gender, and location, which only roughly narrow the search area of users. Still, the program does not have any algorithms for socialization and selection of users by common interests. Another disadvantage is the restriction on the number of users that can be viewed per day. Accordingly, the regulation can be lifted if you buy a paid subscription for 5\$.

How the *Tinder* app works is shown in Fig. 1.



Figure 1: Tinder app screens

*Badoo* is a reasonably popular system for socializing individuals, which is the newest program of all these. The program's main advantage is its presence: it is used as a desktop program on all operating systems, as an application for a smartphone, and in the browser version. It is also worth noting that the program is entirely free respectively. It has no restrictions on the number of users that can be viewed

per day, has an unlimited number of preferences and messages that can be set to users. Among the disadvantages, it is worth noting a considerable number of ads that cannot be turned off, which does not allow you to use the program fully conveniently. Another disadvantage is that *Badoo* does not have any algorithms for socialization based on shared interests and can use only basic user search filters. A screenshot of how *Badoo* works and the main screens of the program are shown in Fig. 2.

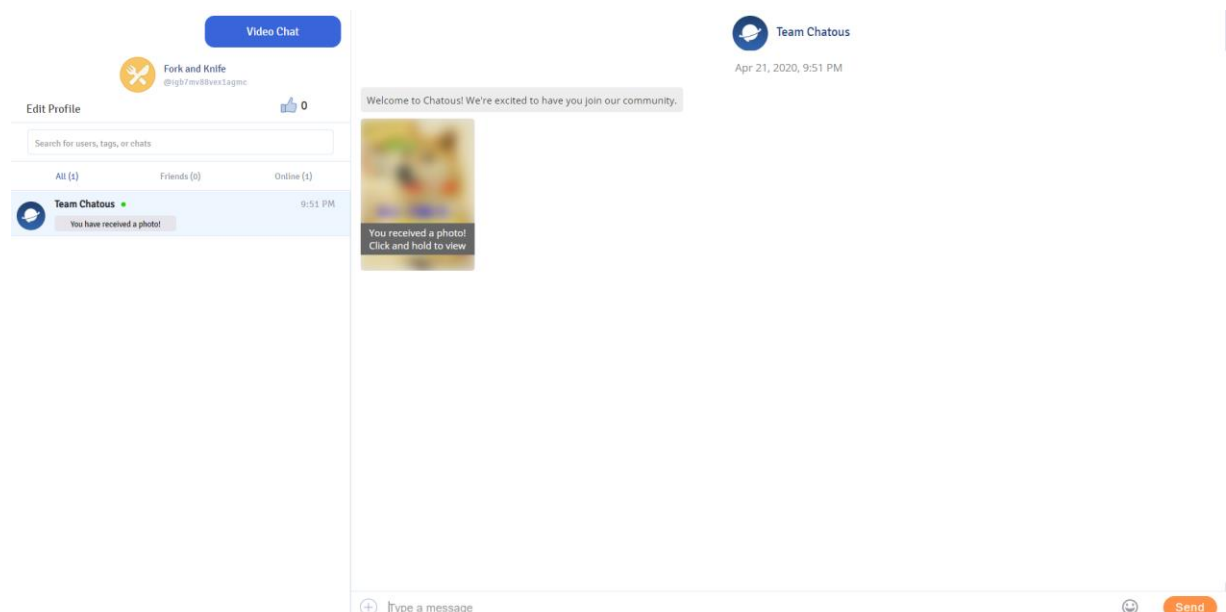


**Figure 2:** Badoo app screens

The last app is *Chatous*, the least popular app of all listed. The main advantage of this program is that it is entirely free to use, has an unlimited number of likes, messages, and user views throughout the day. Also, this program completely lacks any advertising. The advantages include analyzing the user's photos and confirming the user's face in the image, ensuring that we communicate with a real user.

Among the disadvantages is that this program is not available on all platforms, but only as a browser program, limiting the number of possible users of the program. Its main drawback is the absence of algorithms for selecting users by interests, which does not allow you to search for users optimally. There are only basic algorithms for filtering the main user parameters, which is inefficient since most users will not meet search expectations, as they will not have common interests.

A screenshot of how *Chatous* works and the main program screens are shown in Fig. 3.



**Figure 3:** Chatous app screens

All advantages and disadvantages of analogue programs are shown in a comparative Table 1.

**Table 1**

Comparison of analogue programs of the intelligent system

| Name                     | Tinder | Badoo | Chatous |
|--------------------------|--------|-------|---------|
| Lack of advertising      | +      | -     | +       |
| Unlimited users          | -      | +     | +       |
| Unlimited likes          | -      | +     | +       |
| Unlimited messages       | -      | +     | +       |
| Paid version             | +      | -     | -       |
| Cross-platform           | +      | +     | -       |
| Photo analysis           | +      | +     | +       |
| Basic search filters     | +      | +     | +       |
| Socialization algorithms | -      | -     | -       |

After analyzing the data obtained in Table 1, we can conclude that an essential point in creating an intelligent system is the absence of advertising or its minimum number. Also, the advantage over *Tinder* will be an unlimited number of users, likes that can be sent to users, and messages that can be written throughout the day. Also, the advantages of *Tinder* and *Badoo* apps are the availability of apps to work on all platforms, which significantly expands the number of potential users. The main result of the analysis is the algorithms that the systems under study provide. It is necessary to implement a photo analysis algorithm that can determine a human face in the photo and search for similar pictures in the system database. Another critical point is the use of basic user search filters available in all three programs. The main advantage of the user socialization intelligent system based on shared interests should be implementing socialization algorithms, which are not available in all three programs. It will optimize the user search process as much as possible by analyzing the user's interests and essential information and selecting the most relevant users of the system.

### 3. Materials and methods

#### 3.1. System analysis for intelligent system design and development

The main goal and aspects of the system's functioning are formulated using the goal tree to describe the intelligent system. In this case, the system is regarded as an «Intelligent system for the socialization of individuals by common interests (G).» It is designed to store and analyze all system users, processes the input information, compares users' interests, and enhance their further socialization [42-49]. It outlines only a general understanding of what the intelligent system should do. A more detailed explanation of the system's operation is necessary to specify its main aspects [50-53]. The first level includes four main elements: 1) data transfer (A1), 2) data analysis (A2), 3) data storage (A3), and 4) data generation (A4). In this case, all aspects of the first level of a detailed goal tree are presented, which gives a specific idea of the basic algorithm of systems, but it is still not enough to fully formulate the system's features. The second level includes: structuring (B1), functionality (B2), security (B3), integrity (B4), and scalability (B5).

According to the third level, the criteria that are sub-criteria belong to the second level: structurings consist of patterning (B1.1) and hierarchy (B1.2), functionality consists of operability (B2.1) and representation (B2.2), security consists of recoverability (B3.1) and reserve capacity (B3.2), integrity consists of synchronicity (B4.1) and dependency (B4.2), scalability consists of adaptability (B5.1) and mobility (B5.2). In this case, the fourth level of the hierarchy will be the last. It contains two alternatives: an information-analytical system and an information-reference system, indicated in the diagram as alternative 1 (C1) and alternative 2 (C2), among which it is necessary.

Fig. 4 shows a goal tree of the intelligent system containing four levels of hierarchy. The next step is to prioritize hierarchical elements. Values from 1 to 5 were used in the evaluation. One means a similar deal, 3 represents a weak value, 5 indicates a substantial weight, and 2 or 4 means intermediate values between neighbouring elements. Table 2 shows the constructed matrix of pairwise comparisons.



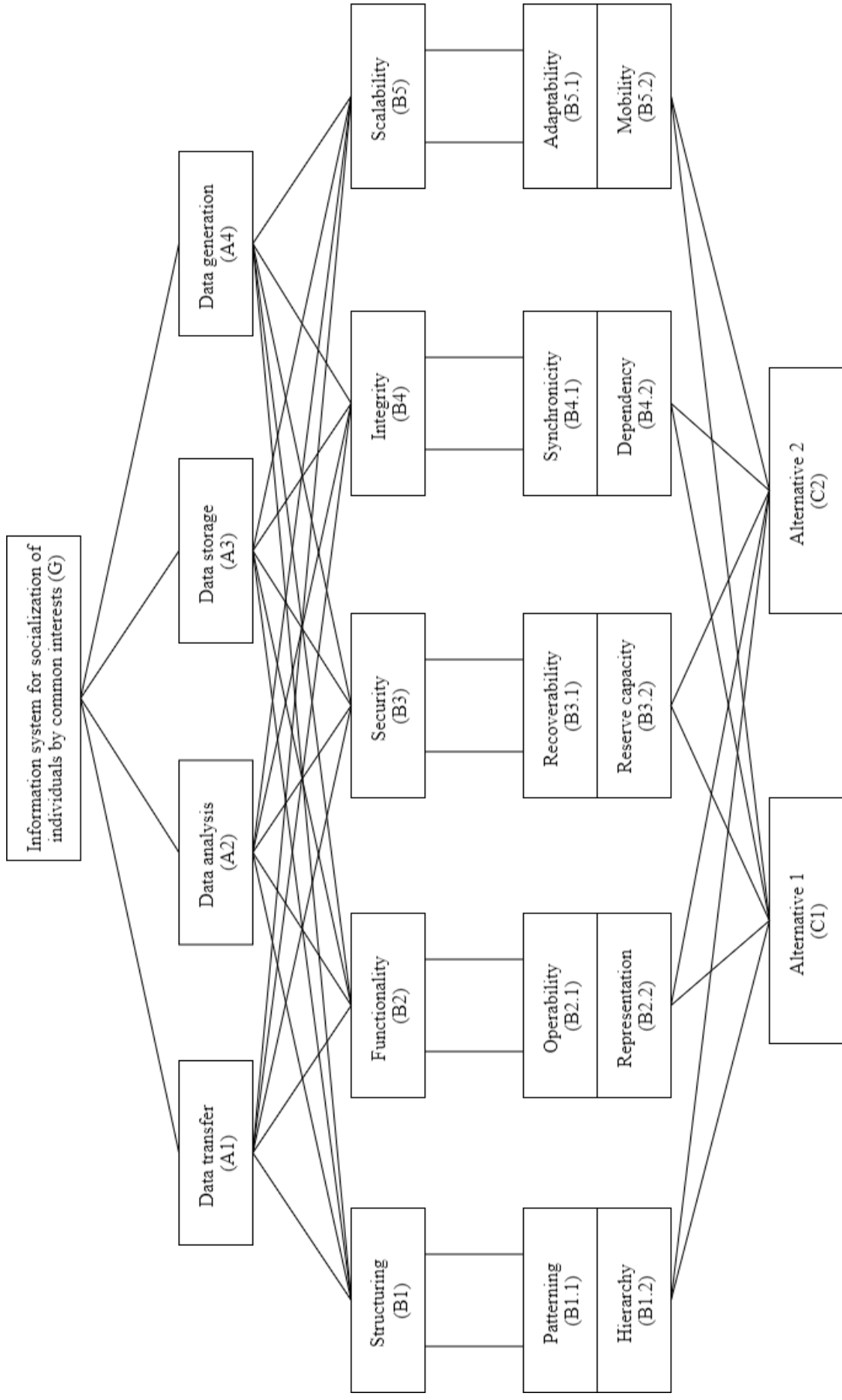


Figure 4: Goal tree of information system

**Table 2**

Matrix of pairwise comparisons of the system goal

| Intelligent system for the socialization of individuals by common interests | Data transfer | Data analysis | Data storage | Data generation |
|---|---------------|---------------|--------------|-----------------|
| Data transfer   | 1             | 2             | 3            | 5               |
| Data analysis   | 0,5           | 1             | 0,5          | 3               |
| Data storage  | 0,333         | 2             | 1            | 4               |
| Data generation   | 0,2           | 0,333         | 0,25         | 1               |

Using the Thomas Saati algorithm, we found the weights of criteria that received the following values: data transfer – 0,357, data analysis – 0,389, data storage – 0,192, data generation - 0,062.

The next step was to find the largest value:  $\lambda_{\max} = 0,357 * (1 + 0,333 + 0,5 + 0,2) + 0,389 * (2 + 0,333 + 2 + 1) + 0,192 * (1 + 0,25 + 3 + 0,5) + 0,062 * (3 + 5 + 1 + 4) = 2,07 + 0,725 + 0,912 + 0,806 = 4,213$ . The consistency index was found  $CI = (4,213 - 4) / 3 = 0,071$  and the sequence index  $CR = 0,071 / 0,90 = 0,078$  was also calculated. It is worth noting that 0.90 is a value taken from the random index table, since the current index we need is 4.

The next step is to build a pairwise comparison matrix to analyze the second level of the hierarchy based on the first level, as shown in Tables 3.

**Table 3**

Matrix of pairwise comparisons of the all criteria's

| Criteria        | Name          | Structuring | Functionality | Security | Integrity | Scalability |
|-----------------|---------------|-------------|---------------|----------|-----------|-------------|
| Data transfer   | Structuring   | 1           | 3             | 2        | 0,5       | 0,333       |
|                 | Functionality | 0,333       | 1             | 1        | 0,333     | 0,2         |
|                 | Security      | 0,5         | 1             | 1        | 0,5       | 0,2         |
|                 | Integrity     | 2           | 3             | 2        | 1         | 1           |
|                 | Scalability   | 3           | 5             | 5        | 1         | 1           |
| Data analysis   | Structuring   | 1           | 4             | 2        | 3         | 1           |
|                 | Functionality | 0,25        | 1             | 1        | 0,5       | 0,2         |
|                 | Security      | 0,5         | 1             | 1        | 0,333     | 0,2         |
|                 | Integrity     | 0,333       | 2             | 3        | 1         | 0,333       |
|                 | Scalability   | 1           | 5             | 5        | 3         | 1           |
| Data storage    | Structuring   | 1           | 3             | 5        | 3         | 0,333       |
|                 | Functionality | 0,333       | 1             | 2        | 1         | 0,2         |
|                 | Security      | 0,2         | 0,5           | 1        | 3         | 0,25        |
|                 | Integrity     | 0,333       | 1             | 0,333    | 1         | 0,5         |
|                 | Scalability   | 3           | 5             | 4        | 2         | 1           |
| Data generation | Structuring   | 1           | 2             | 3        | 3         | 2           |
|                 | Functionality | 0,5         | 1             | 3        | 2         | 1           |
|                 | Security      | 0,333       | 0,333         | 1        | 0,5       | 0,333       |
|                 | Integrity     | 0,333       | 0,5           | 2        | 1         | 3           |
|                 | Scalability   | 0,5         | 1             | 3        | 0,333     | 1           |

After performing calculations taken from matrices of pairwise comparisons of system criteria similar to those performed at level 1 of the hierarchy, the following priority values were obtained for level 2 of the hierarchy. For data transfer: structuring – 0,164, functionality – 0,182, security – 0,255, integrity – 0,089, scalability – 0,310. For data analysis: structuring – 0,164, functionality – 0,182, security – 0,255, integrity – 0,089, scalability – 0,310. For data storage: structuring – 0,164, functionality – 0,182, security – 0,255, integrity – 0,089, scalability – 0,310. For data generation: structuring – 0,164, functionality – 0,182, security – 0,255, integrity – 0,089, scalability – 0,310.

Next, global priorities were calculated according to the criteria of the 2nd hierarchy level and their sub-criteria obtained from the 3rd level. The main criteria were selected: structure, functionality, security, integrity, and scalability. A matrix of pairwise comparisons for the third level of the hierarchy relative to the second level concerning all criteria is created and shown in Table 4. After performing all the necessary calculations, the values of global priorities shown in Table 5 were obtained. Then the consistency check was completed, the importance of all  $\lambda_{max}$ , CI, CR were found, and as a result, it was proved that all CR values do not exceed the limit of 10% (Table 5). The last step was to build matrices according to the size of available alternatives, where comparisons are made based on level 3 criteria. The resulting matrices are shown in Table 6.

**Table 4**  
Matrix of pairwise comparisons of the *structuring* criteria

|                  |                    |                     |
|------------------|--------------------|---------------------|
| Structuring      | Patterning         | Hierarchy           |
| Patterning       | 1                  | 3                   |
| Hierarchy        | 0,333              | 1                   |
| Functionality    | Operability        | Representation      |
| Operability      | 1                  | 2                   |
| Representation   | 0,5                | 1                   |
| Security         | Recoverabilit<br>y | Reserve<br>capacity |
| Recoverability   | 1                  | 0,2                 |
| Reserve capacity | 5                  | 1                   |
| Integrity        | Synchronicity      | Dependency          |
| Synchronicity    | 1                  | 5                   |
| Dependency       | 0,2                | 1                   |
| Scalability      | Adaptability       | Mobility            |
| Adaptability     | 1                  | 3                   |
| Mobility         | 0,333              | 1                   |

**Table 5**  
Global priorities obtained

| Criteria                 | Name             | Value | $\lambda_{max}$ | CI    | CR    |
|--------------------------|------------------|-------|-----------------|-------|-------|
| Structuring<br>(0,188)   | Patterning       | 0,090 | 2,034           | 0,055 | 0,071 |
|                          | Hierarchy        | 0,098 |                 |       |       |
| Functionality<br>(0,092) | Operability      | 0,029 | 2,065           | 0,045 | 0,065 |
|                          | Representation   | 0,063 |                 |       |       |
| Security<br>(0,322)      | Recoverability   | 0,277 | 2,080           | 0,049 | 0,063 |
|                          | Reserve capacity | 0,045 |                 |       |       |
| Integrity<br>(0,121)     | Synchronicity    | 0,076 | 2,043           | 0,052 | 0,072 |
|                          | Dependency       | 0,045 |                 |       |       |
| Scalability<br>(0,277)   | Adaptability     | 0,192 | 2,011           | 0,058 | 0,075 |
|                          | Mobility         | 0,085 |                 |       |       |

**Table 6**

Matrix of pairwise comparisons of the all criteria's

|                         | A1    | A2   | Alternative weight |
|-------------------------|-------|------|--------------------|
| <b>Patterning</b>       |       |      |                    |
| A1                      | 1     | 3    | 0,75               |
| A2                      | 0,333 | 1    | 0,25               |
| <b>Hierarchy</b>        |       |      |                    |
| A1                      | 1     | 0,25 | 0,2                |
| A2                      | 4     | 1    | 0,8                |
| <b>Operability</b>      |       |      |                    |
| A1                      | 1     | 1    | 0,5                |
| A2                      | 1     | 1    | 0,5                |
| <b>Representation</b>   |       |      |                    |
| A1                      | 1     | 1    | 0,5                |
| A2                      | 1     | 1    | 0,5                |
| <b>Recoverability</b>   |       |      |                    |
| A1                      | 1     | 0,25 | 0,2                |
| A2                      | 4     | 1    | 0,8                |
| <b>Reserve capacity</b> |       |      |                    |
| A1                      | 1     | 5    | 0,83               |
| A2                      | 0,2   | 1    | 0,17               |
| <b>Synchronicity</b>    |       |      |                    |
| A1                      | 1     | 1    | 0,5                |
| A2                      | 1     | 1    | 0,5                |
| <b>Dependency</b>       |       |      |                    |
| A1                      | 1     | 2    | 0,75               |
| A2                      | 0,333 | 1    | 0,25               |
| <b>Adaptability</b>     |       |      |                    |
| A1                      | 1     | 0,25 | 0,2                |
| A2                      | 4     | 1    | 0,8                |
| <b>Mobility</b>         |       |      |                    |
| A1                      | 1     | 3    | 0,75               |
| A2                      | 0,333 | 1    | 0,25               |

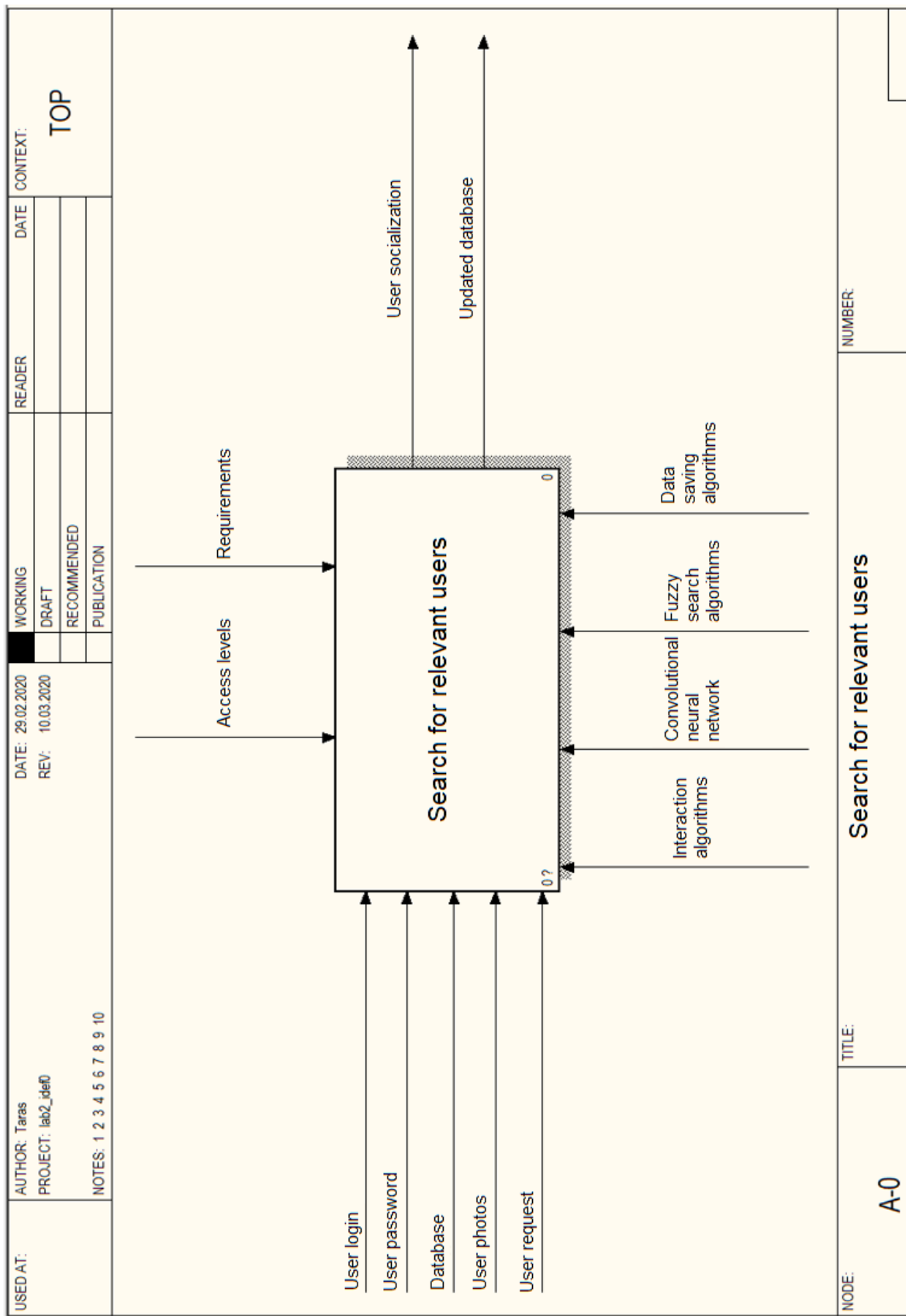
After analyzing the matrices of pairwise comparisons of the criteria of the last level of the hierarchy relative to the two current existing criteria of the Intelligent System, the weights of the current alternatives to each criterion were obtained, the global weights were calculated as the sum of the products of the data of global values for the second level of the hierarchy, namely: structuring, functionality, security, integrity, and scalability on the values for the current criteria taken from the third level of the hierarchy, namely: patterning, hierarchy, operability, representation, recoverability, reserve capacity, synchronicity, dependency, adaptability, and mobility.

Alternatives have the following values: A1 (intelligent-analytical system) = 0.602, A2 (information-reference system) = 0.398. In conclusion, we can say that the intelligent-analytical system is the best alternative according to all criteria.

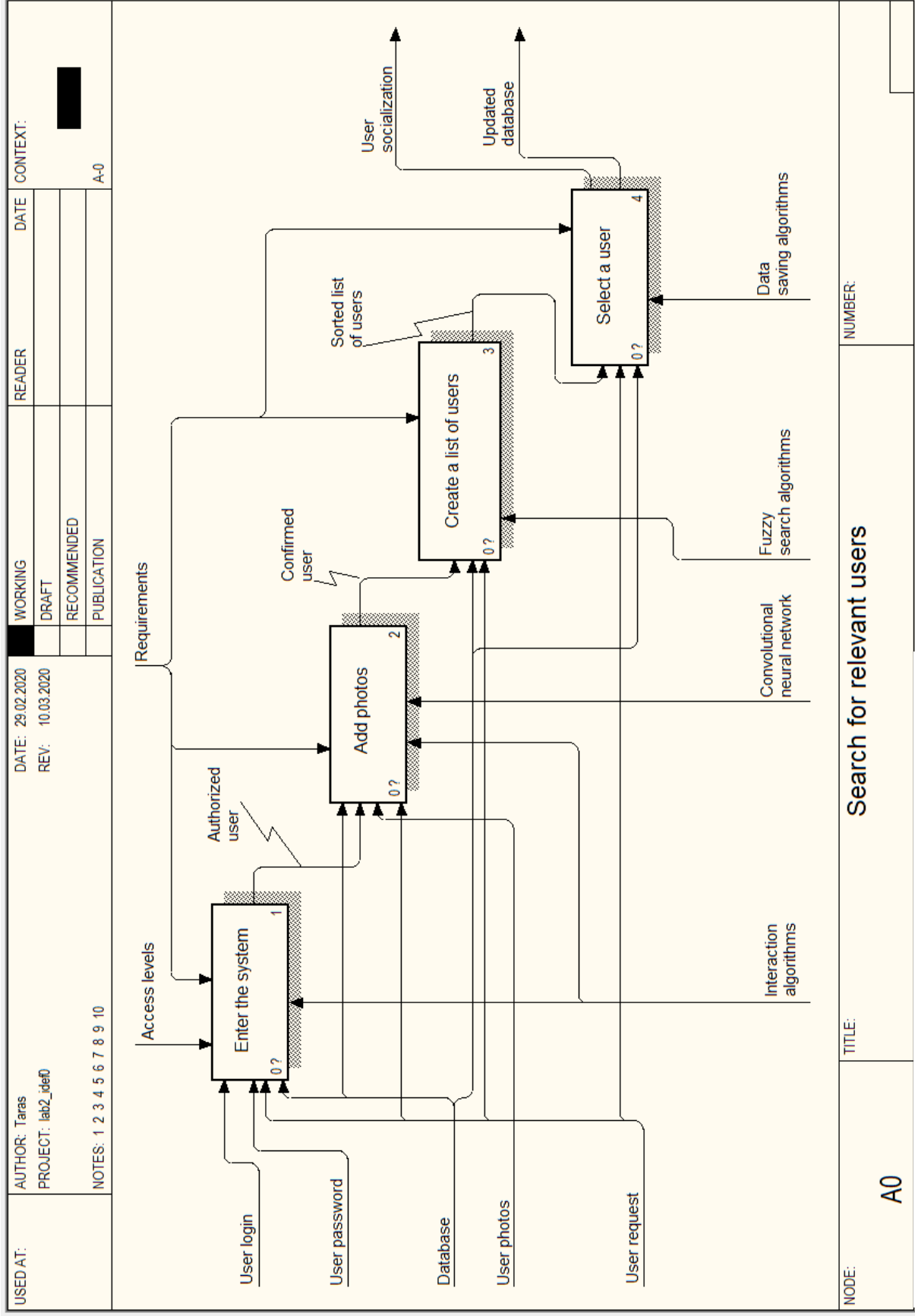
### 3.2. Specification of the system's functioning

After constructing the goal tree and related works [21-49], the system functions were further specified, and IDEF0 functional diagrams were selected to detail the system structure (Fig. 5-9).

First of all, to generalize the main functions of the created intelligent system, a contextual diagram was created, which consists of the main input and output data and primary data for displaying mechanisms and controls. Input data includes user login, user password, database, user photos, and user request. Initial aspects include user socialization and an updated database.



**Figure 5:** Context diagram of the intelligent system

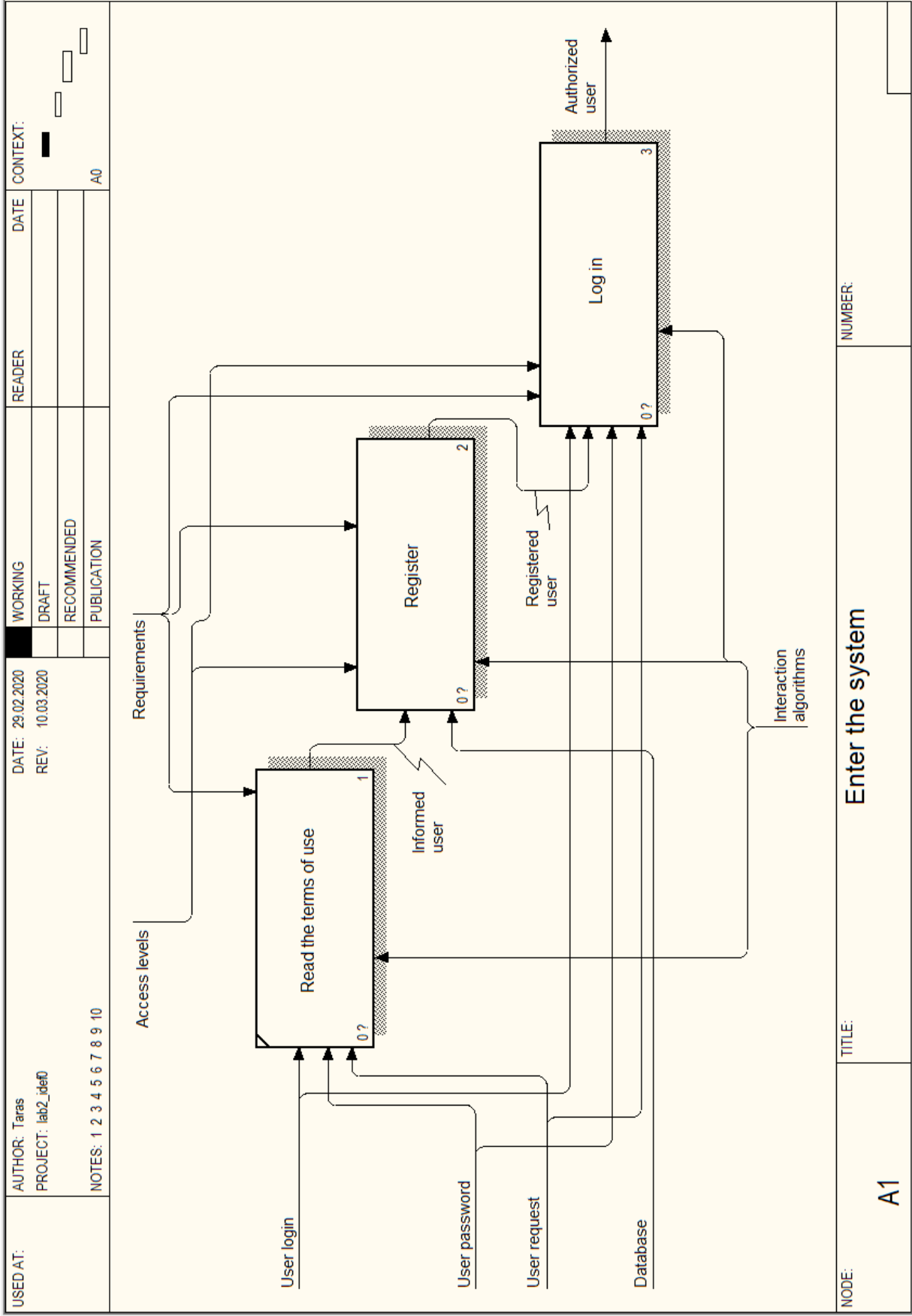


**Search for relevant users**

NOTE: TITLE: A0

NUMBER:

**Figure 6:** Intelligent system decomposition



NODE: A1

TITLE: Enter the system

NUMBER:

Figure 7: Decomposition of the "Enter the system" block

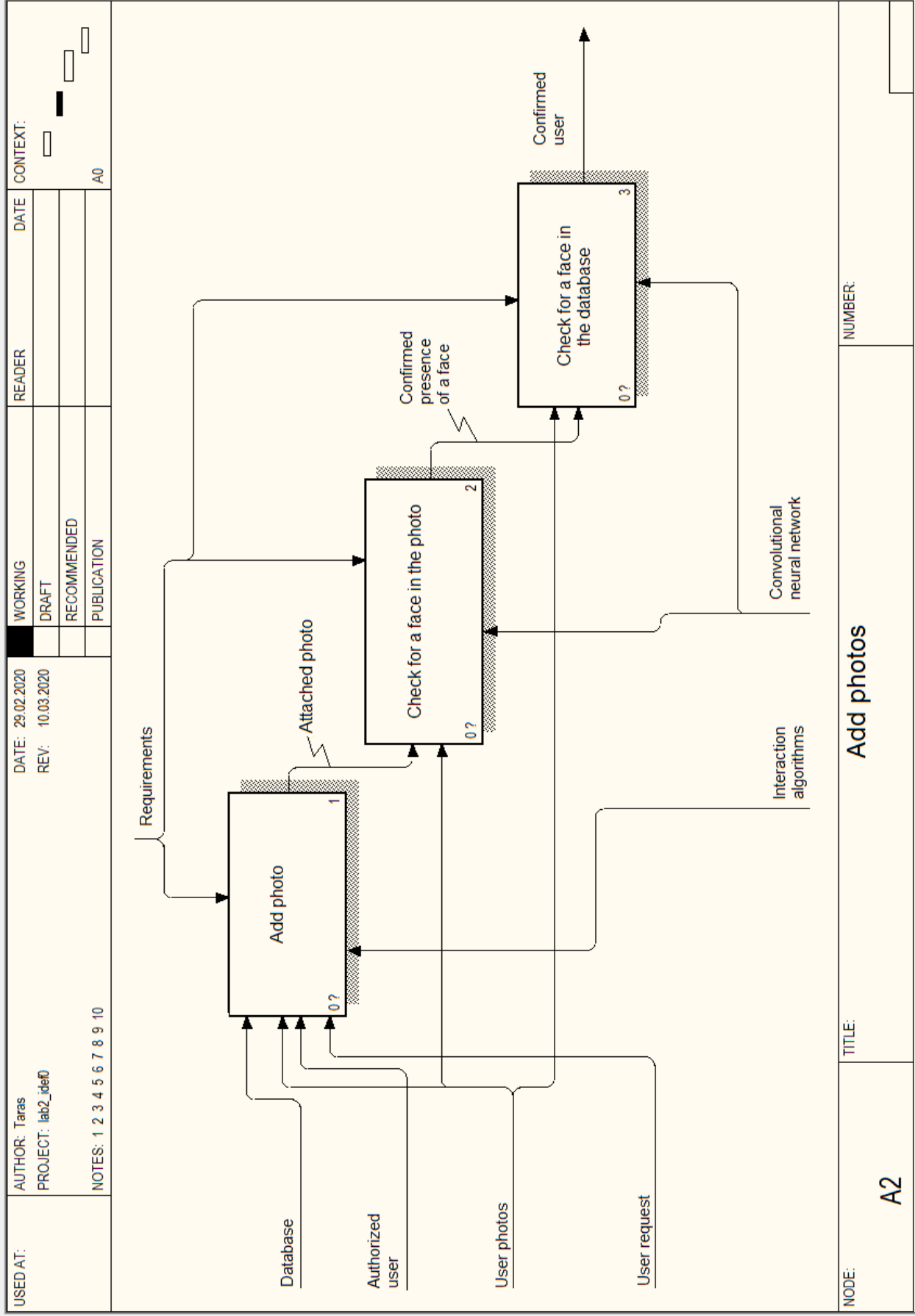
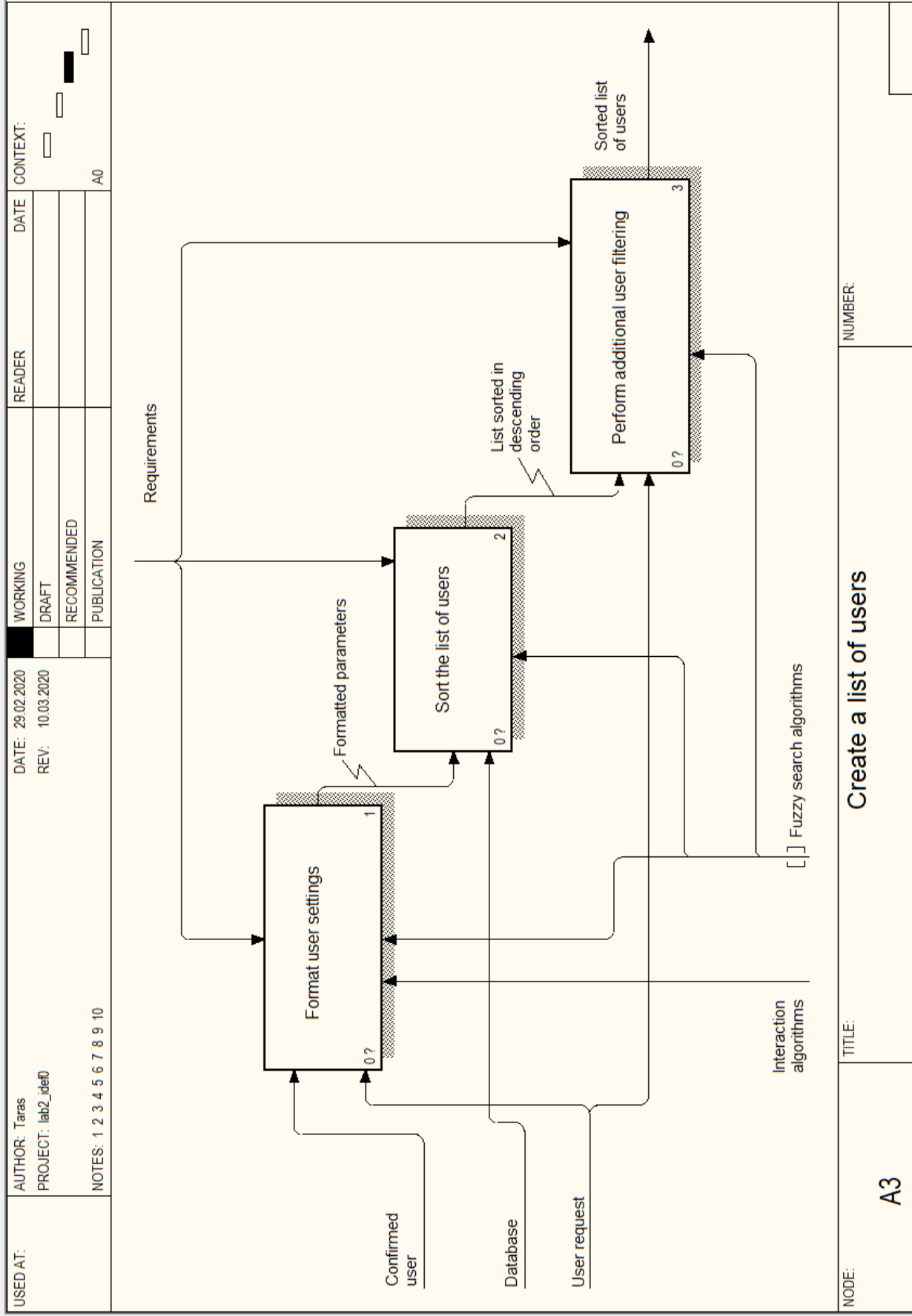


Figure 8: Decomposition of the "Add Photos" block





**Figure 9:** Decomposition of the “Create a list of users” block

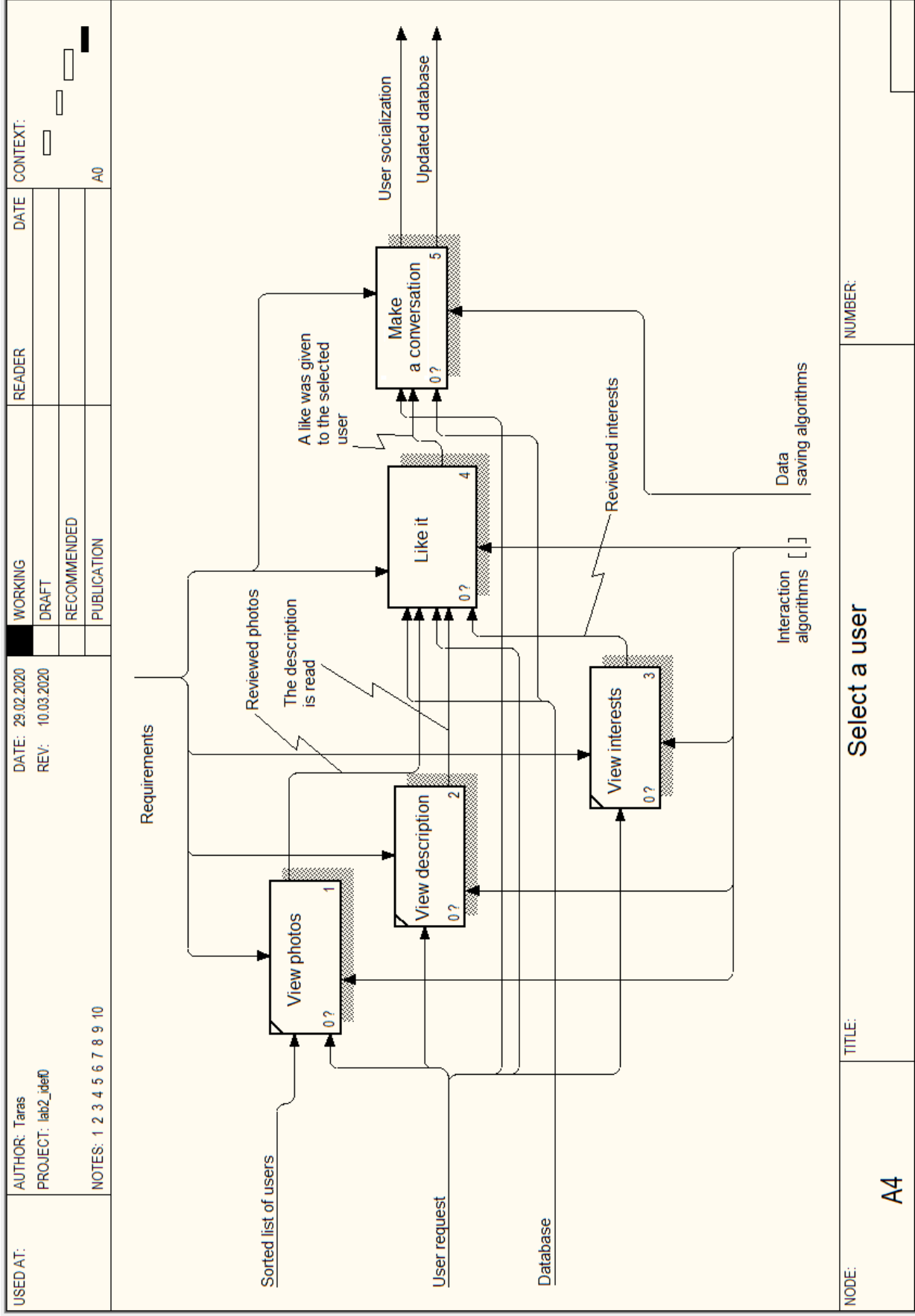
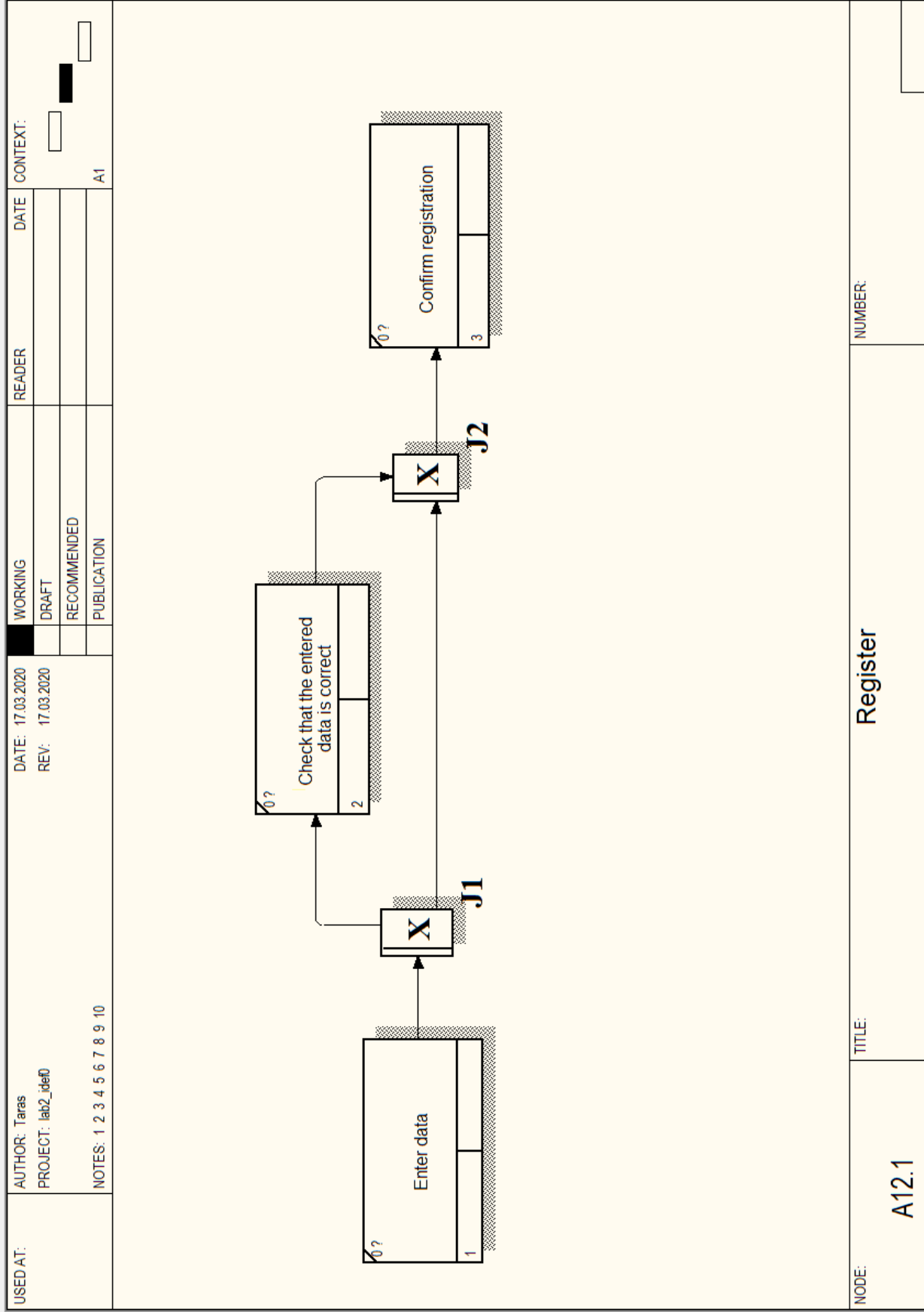
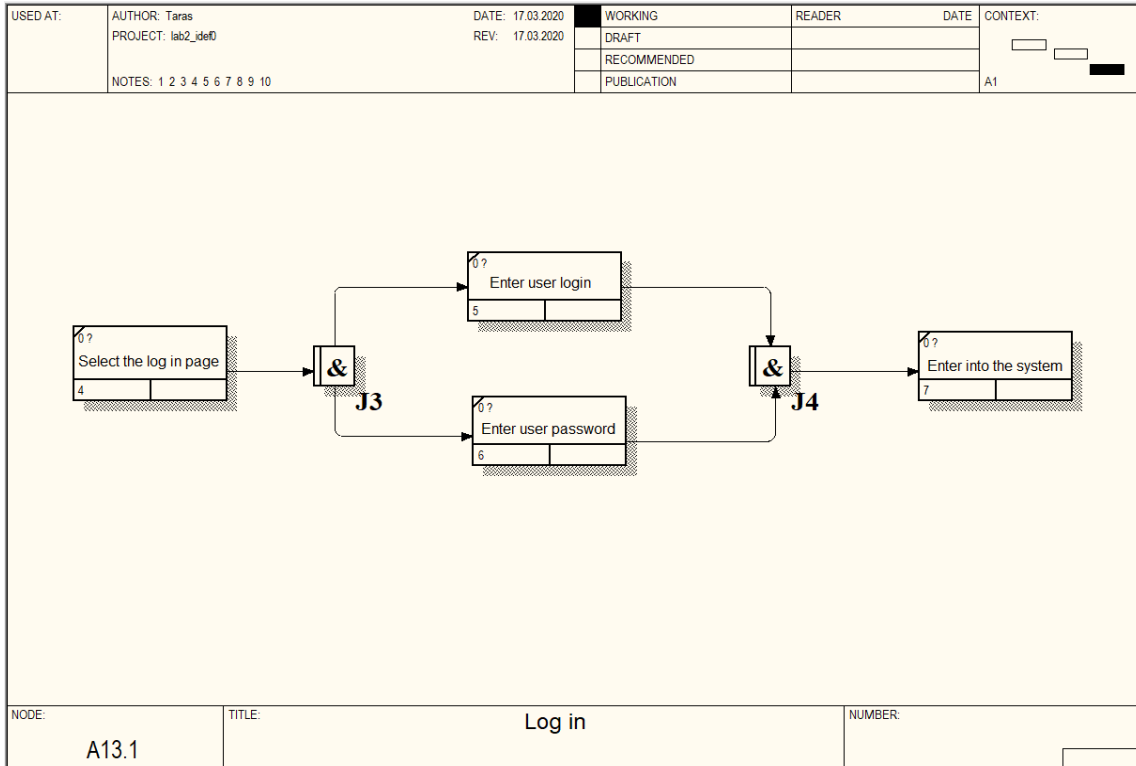


Figure 10: Decomposition of the "Select a user" block

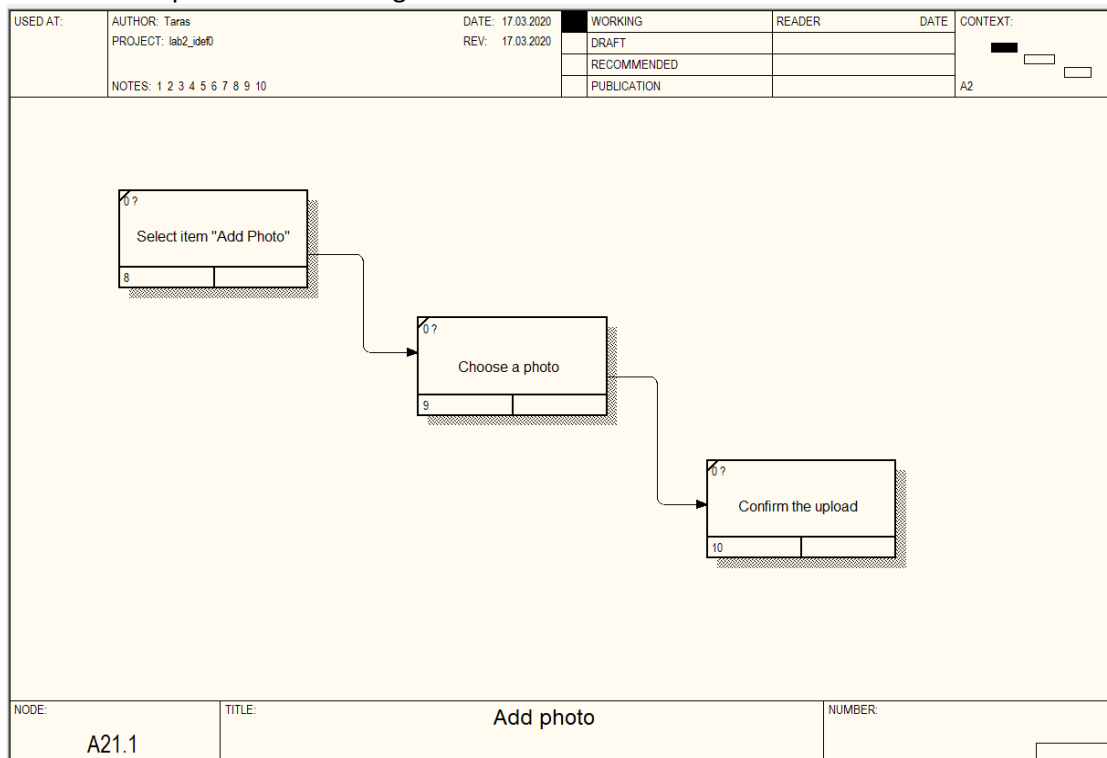


**Figure 11:** Decomposition of the "Register" block

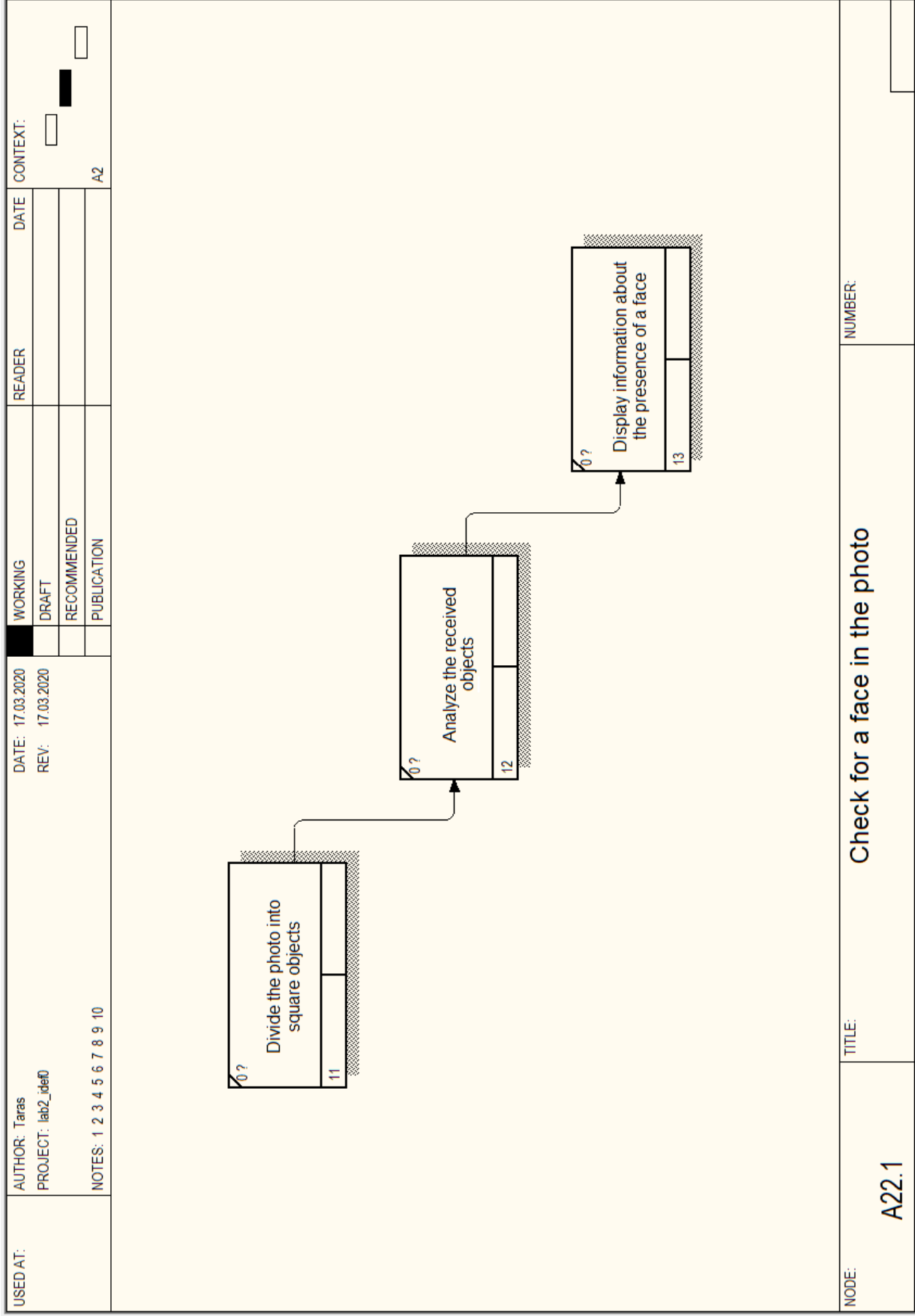
Among the main elements of control, it is necessary to highlight access levels and requirements that affect the processes of the intelligent system. The system's primary functions are performed by the system's mechanisms: interaction algorithms, convolutional neural network, fuzzy search algorithms, and data saving algorithms. Fig. 12 shows the decomposition of the *Log in* block. Fig. 13-15 show the decomposition of the "Add photo," "Check for a face in the photo," and "Check for a face in the database" blocks.



**Figure 12:** Decomposition of the "Log in" block

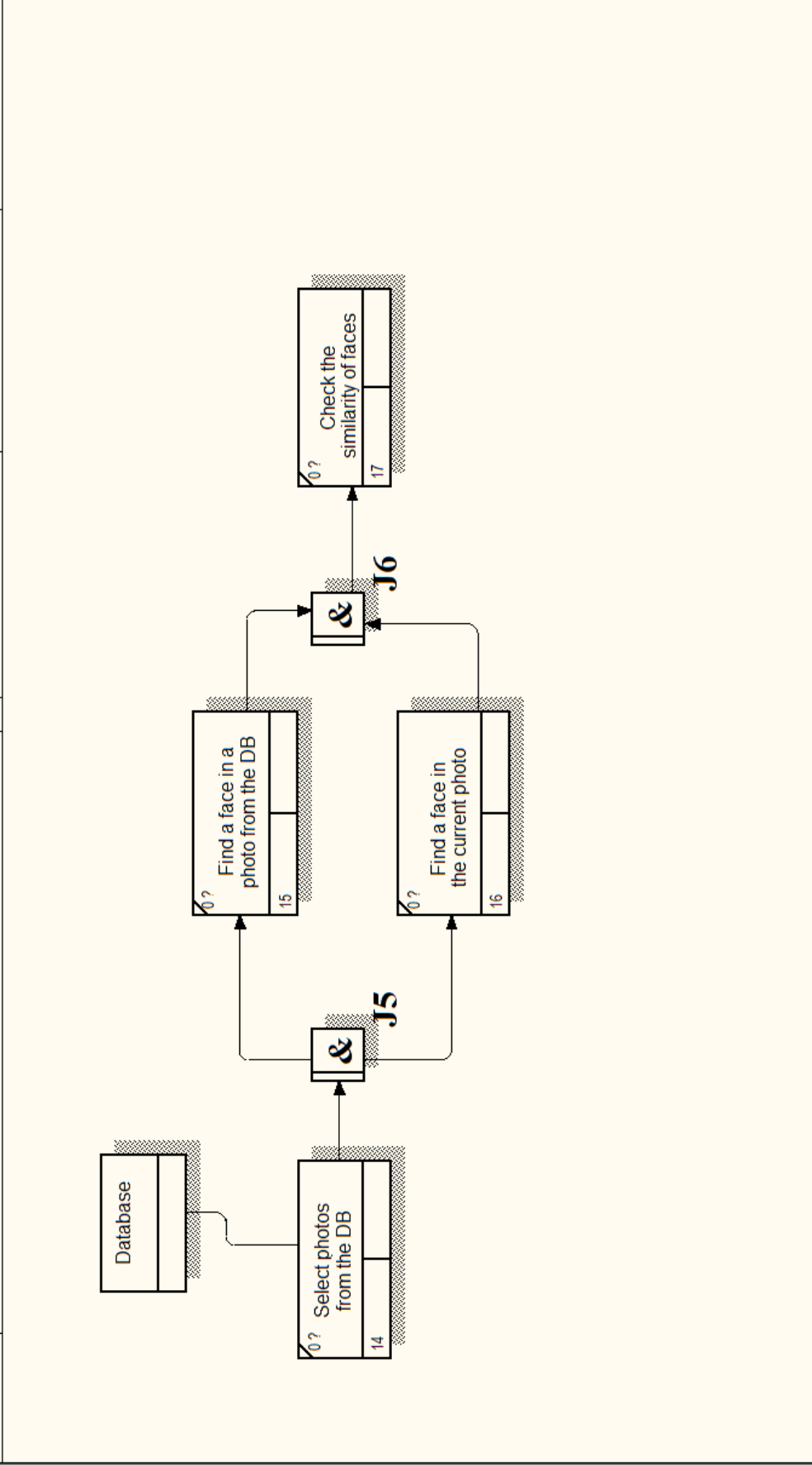


**Figure 13:** Decomposition of the "Add photo" block



**Figure 14:** Decomposition of the “Check for a face in the photo” block

|          |                             |                  |             |        |      |          |
|----------|-----------------------------|------------------|-------------|--------|------|----------|
| USED AT: | AUTHOR: Taras               | DATE: 17.03.2020 | WORKING     | READER | DATE | CONTEXT: |
|          | PROJECT: lab2_def0          | REV: 17.03.2020  | DRAFT       |        |      |          |
|          |                             |                  | RECOMMENDED |        |      |          |
|          | NOTES: 1 2 3 4 5 6 7 8 9 10 |                  | PUBLICATION |        |      | A2       |



|       |       |        |                                  |         |  |
|-------|-------|--------|----------------------------------|---------|--|
| NODE: | A23.1 | TITLE: | Check for a face in the database | NUMBER: |  |
|-------|-------|--------|----------------------------------|---------|--|

Figure 15: Decomposition of the "Check for a face in the database" block

Fig. 16 shows the decomposition of the "Format user" settings block. Fig. 17-18 show the decomposition of the "Sort the list of users" and "Perform additional user filtering" blocks.

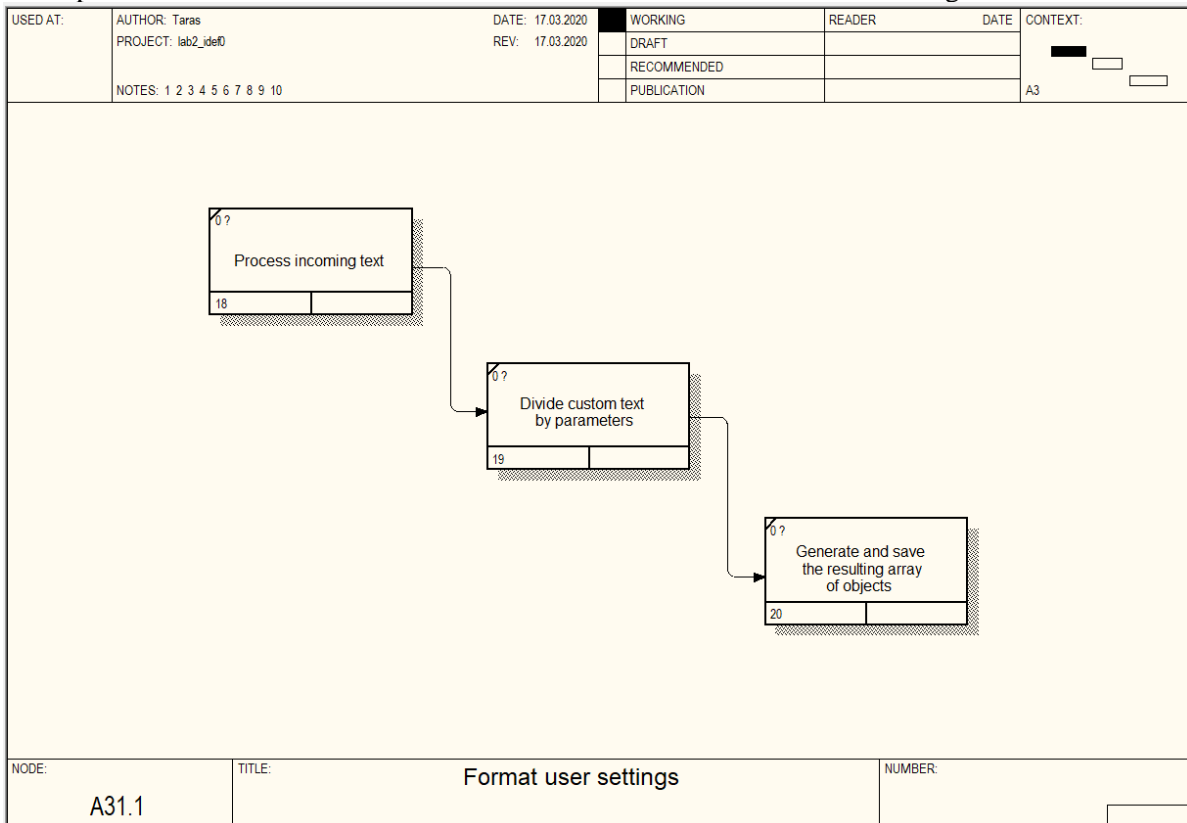


Figure 16: Decomposition of the "Format user settings" block

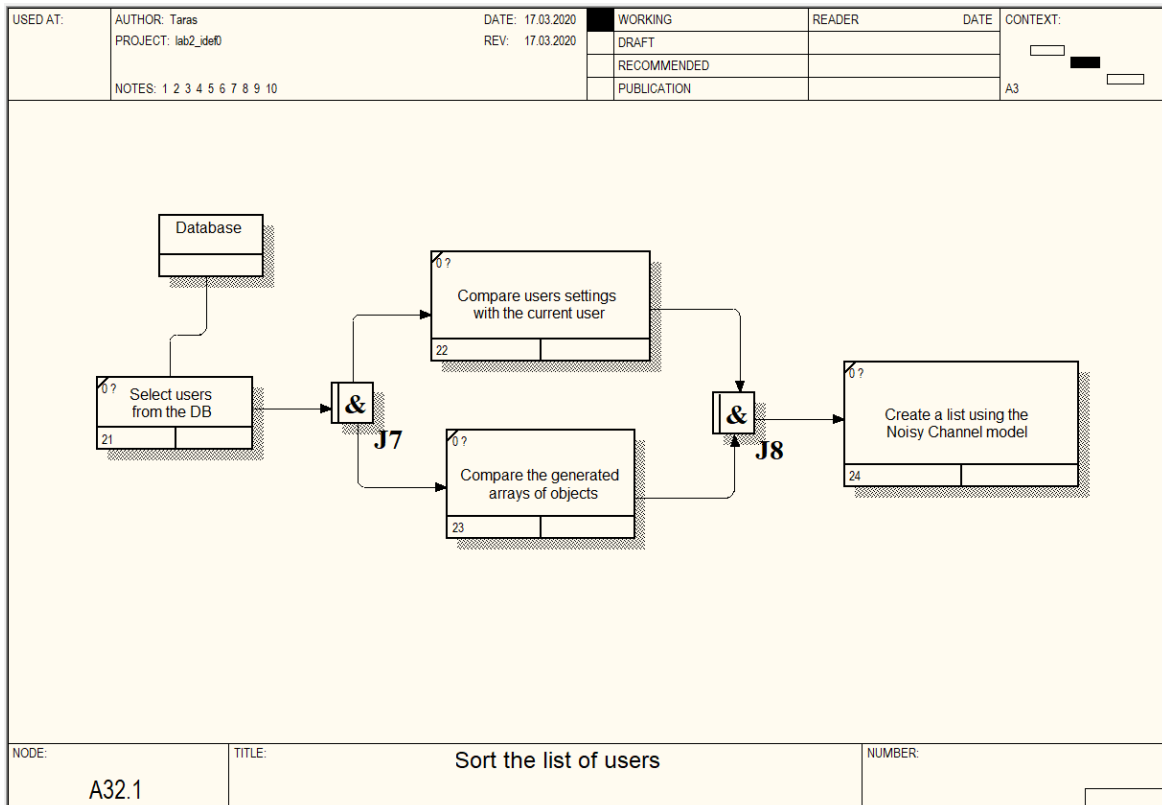
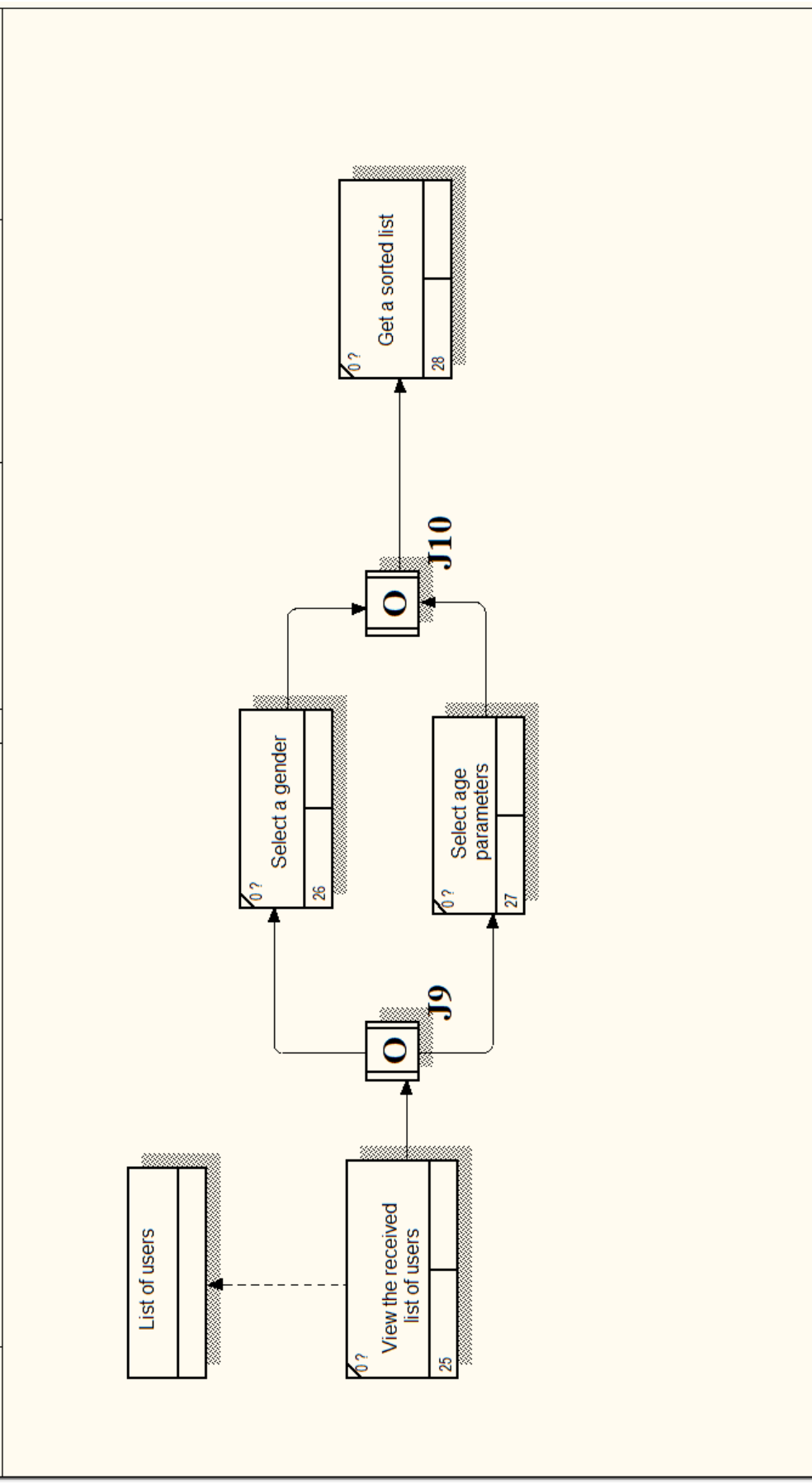


Figure 17: Decomposition of the "Sort the list of users" block

|          |                    |                  |             |        |      |          |
|----------|--------------------|------------------|-------------|--------|------|----------|
| USED AT: | AUTHOR: Taras      | DATE: 17.03.2020 | WORKING     | READER | DATE | CONTEXT: |
|          | PROJECT: lab2_ide0 | REV: 17.03.2020  | DRAFT       |        |      |          |
|          |                    |                  | RECOMMENDED |        |      |          |
|          |                    |                  | PUBLICATION |        |      | A3       |

NOTES: 1 2 3 4 5 6 7 8 9 10

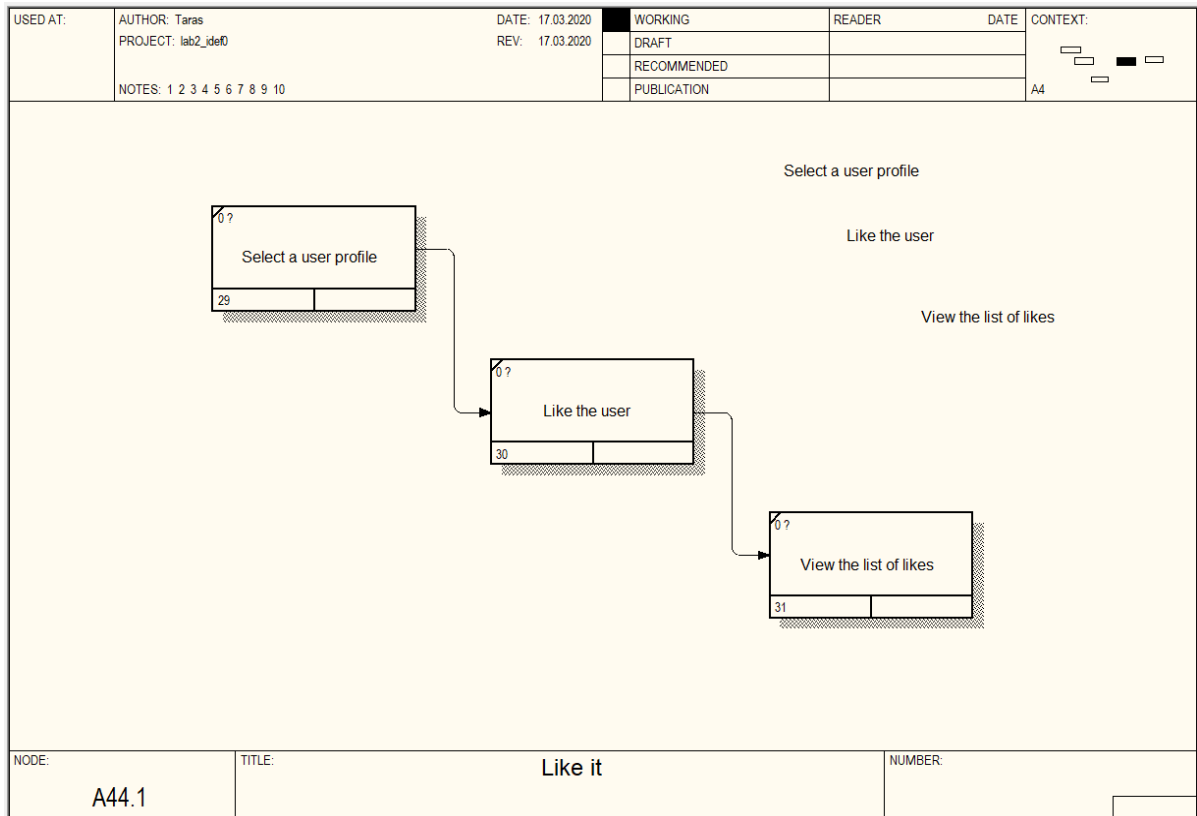


|             |  |         |
|-------------|--|---------|
| NODE: A33.1 | TITLE: Perform additional user filtering | NUMBER: |
|-------------|--|---------|

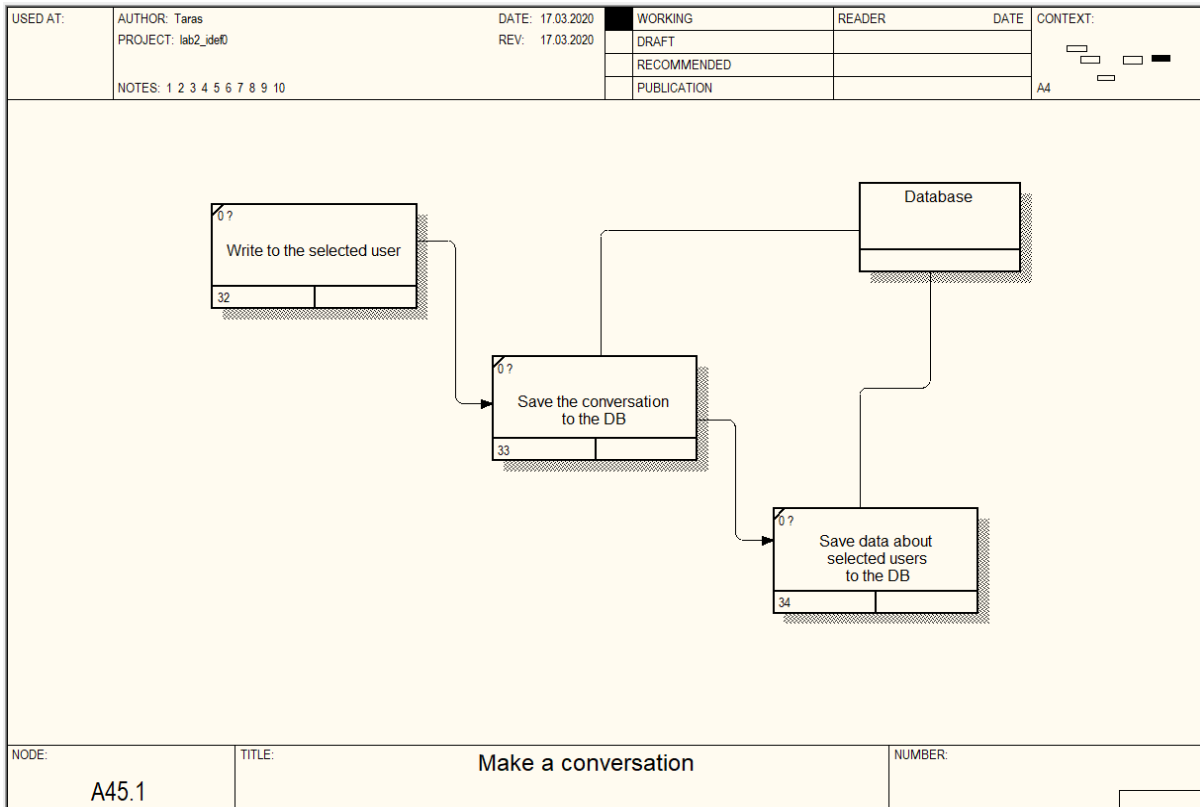
Figure 18: Decomposition of the "Perform additional user filtering" block



Fig. 19 shows the decomposition of the "Like it" block. Fig. 20 shows the decomposition of the "Make a conversation" block.



**Figure 19:** Decomposition of the "Like it" block



**Figure 20:** Decomposition of the "Make a conversation" block

### **3.3. Building a hierarchy of processes (functions, tasks)**

A tree-like display structure is used to display a transparent form of the intelligent system and the order of processes since it is the easiest way to represent a clear hierarchy of system elements. The primary root process of an intelligent strategy is "Search for relevant users," as this is the most generalized explanation of all the processes that occur during the system's operation. The root element is divided into four specific parts, namely "Enter the system," "Add photos," "Create a list of users," and "Select a user." Each of these four processes performs its unique functions, taking only specific data for input and sending the generated data samples for output. If we continue the hierarchy of methods, each of these processes consists of a certain number of elements. The "Enter the system" process consists of the main aspects that are responsible for the initial user interaction with the system: "Read the terms of use," "Register," "Log in." The "Add photos" process includes processes that are responsible for adding and processing user photos: "Add photo," "Check for a face in the photo," "Check for a face in the database." The following necessary process, "Create a list of users," includes techniques that are responsible for selecting suitable users of the system: "Format user settings," "Sort the list of users," "Perform additional user filtering." The "Select a user" process includes processes that are responsible for user socialization: "View photos," "View description," "View interests," "Like it," and "Make a conversation." The hierarchy of processes is shown in Fig. 21. While implementing a system analysis of the intelligent system for the socialization of individuals by common interests, a goal tree of the system was constructed, and the system structure was detailed using functional IDEF0 diagrams. Decomposition was performed, and a hierarchy of processes at the lower levels of the system was constructed. In addition, most of the intelligent system processes were specified using workflow diagrams. A process hierarchy was also created to display the hierarchy levels and their elements.

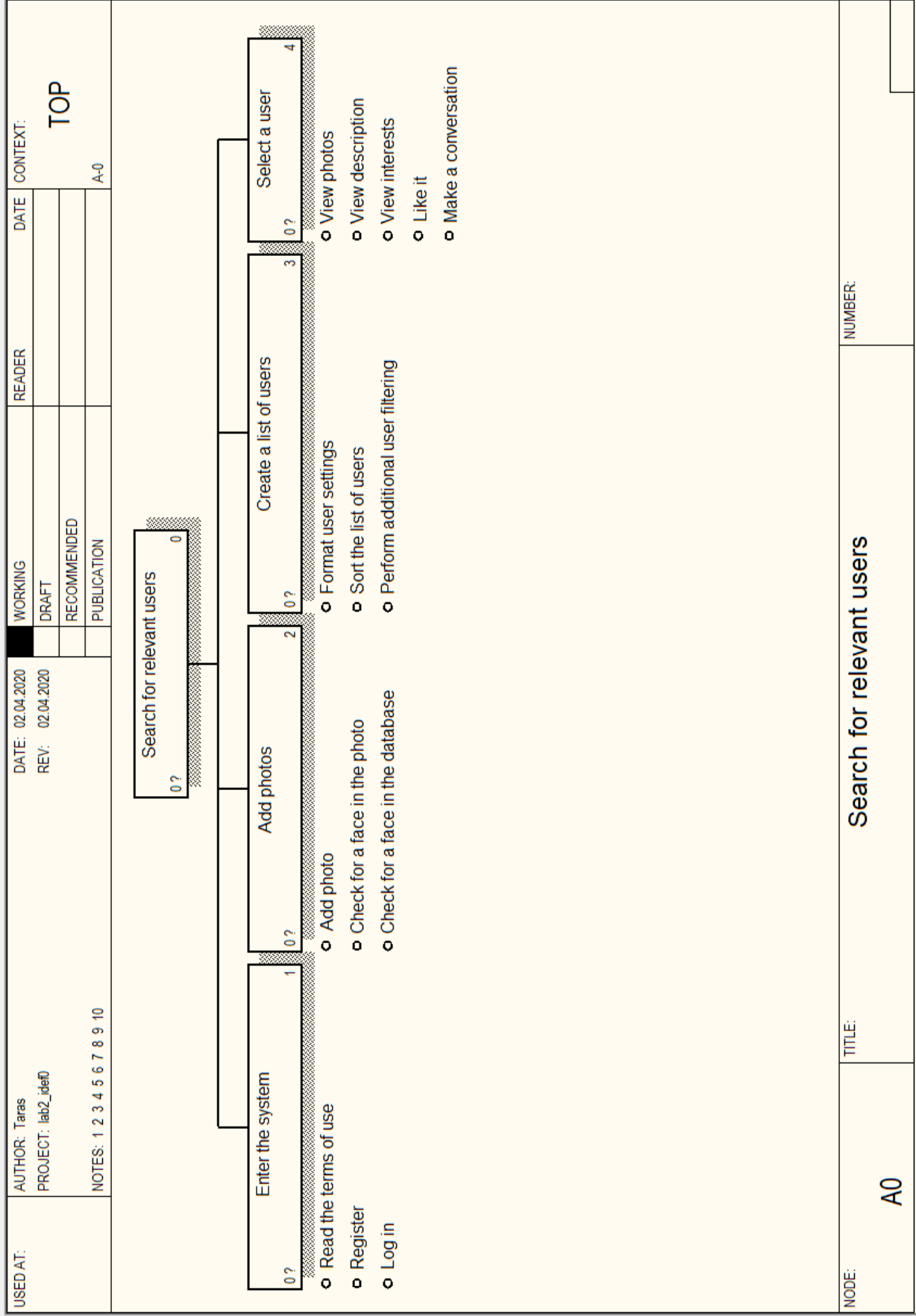
## **4. Experiments**

### **4.1. Selection and justification of tools for solving the problem**

Considering technical characteristics of the C# programming language, it is created as a language with strict typing and not compiled into bytecode. It is compiled into intermediate code that works in a regular extended language environment with other languages from the .NET family of programming languages. The C# programming language is considered a high-level language. Respectively, it has encapsulated many previously popular programming tools, neglecting program execution speed and speeding up the development of software products and creating a convenient environment for programmers. Because of that, they can write structured code, which can then be easily updated and expanded according to the system's requirements. In this case, the C# language is the best choice for writing multi-level code, for implementing the current intelligent approach for the socialization of individuals by common interests.

If we talk about JavaScript, it is a rather specific and difficult-to-use programming language for the web development of client applications. It is very convenient for fast and efficient code writing since JavaScript is a weakly typed programming language. Programmers can neglect the main typing tools and avoid some difficulties and errors during the development of a software system. On the other hand, weak typing can lead to errors in the created system during the execution of tasks.

It is also worth mentioning the TypeScript programming language, which is an add-on to JavaScript. It uses all its essential properties, extends, and complements all aspects of writing software products. Thus, TypeScript is an OOP programming language that provides the programmer with all the vital tools in writing structured code divided into classes and creates the interaction of all classes under the requirements of an intelligent system. You can also use all modern collections to work with data, analyze it, process it, and transform it. It is also worth mentioning the frameworks used, as ASP .NET Core makes it possible to effectively write web applications using all the tools provided for this purpose. The programmer can write a code in the OOP style and using dependency allocation. It allows him to divide the program into models, controllers, and display tools. It attaches all available services inside the system and installs dependencies using existing methods for adding dependencies within one limited Scope component of this program.



**Figure 21:** Hierarchy of intelligent system

Since the intelligent system for the socialization of individuals by common interests is a system with many parts, this approach to structuring all program elements within one project and their interaction is practical. Angular was chosen among the client frameworks, as although it is pretty challenging to learn and use, it is efficient and convenient for writing software products. The framework divides your system into a certain number of components depending on your needs. It uses the tools provided to implement their interaction, which is convenient since you can easily create adjacent system elements and configure everything according to the requirements. You can also implement asynchronous programming, which will allow you to write efficient code while avoiding lengthy and heavy queries that slow down the system and limit functionality. It was necessary to select both the programming language for server development and programming tools and all derivatives for client development during the selection of software tools. Modern software methods and tools were chosen that can be used to create working intelligent systems using all the means provided in programming languages. It was also selected application software that conveniently makes software products, avoiding most of the existing difficulties and problems during creation.

## **4.2. Technical characteristics of the selected software development tools**

To develop a software product, you need to use the software tools that are best suited for creating it. To create a system for socializing individuals by common interests, the type of web application was selected to attract as many users as possible. Creating a technically correct and working web application requires using the right software and appropriate application software for optimal system performance.

First of all, you need to determine the effectiveness of the C# language. It was chosen compared to other server-side programming languages such as Java, Node.js, and others. It is easy to use, modern, has many effective code creation templates and is constantly updated. C# is also a current typed programming language, and respectively, it has a clear division into data types. It is efficient to use fast assembly and compilation of code "on the Go," which allows you to significantly speed up creating a software product and effectively allocate your time. C# has all the modern software tools, namely multi-flow programming, asynchronous code writing, all the necessary data types, and many additional libraries that extend the existing functionality. Also, the advantages of the C# language include simplified code writing and intelligent variables that allow you to avoid explicitly specifying the data type of the selected variable. You can conveniently and quickly use mechanisms for writing asynchronous code using keywords, thereby forming a single code stack. It is also worth highlighting convenient work with files, an intelligent garbage collector, respectively. You do not need to allocate and clean up resources manually. The compiler does it for you. To create a server software product, the C# framework is used, namely ASP .NET Core. This framework uses the MVC programming pattern, i.e., Model, View, and Controller. First of all, this pattern is dominant among all software products due to its versatility and uniqueness. It explicitly divides the application into three layers: the data layer, the data management layer, and the generated data display layer. ASP .NET Core provides all the tools for creating effective web programs. For example, you can automatically create templates for controllers and display files, thereby saving time on writing monotonous code and focusing on writing services that are the basis of the program. Also, ASP .NET Core gives all the necessary tools for effective program creation, and it is possible to create registration and authorization mechanisms using Identity tools conveniently. The framework also provides tools for derivative processing of all processes of the Middleware software system, which is highly convenient in distributing the importance of processes and finding errors since you can throw exceptions in a specific different context. It conveniently processes almost all errors and exceptions that would previously be complex and inefficient in processing and analysis. It is worth highlighting a particular case of the framework, namely the ASP .NET Core Web API, which allows to divide into server and client programs and use cross-domain tools to perform asynchronous requests and send messages. It enables ASP .NET Core to combine convenient client frameworks with writing a multifunctional program using specially dedicated tools for writing software products and systems.

We want to distinguish a unique library of the C# programming language-ML.NET. It effectively uses machine learning tools and provides equal opportunities for creating multifunctional neural networks necessary to use when creating an intelligent system for socializing individuals by common

interests. The library provides a user interface for already made primary neural networks, which can be added using extensions or write your implementations based on existing code. There are many ways to create simple or complex neural networks. One of its main advantages is the convenient documentation of the library, which helps to understand and use all the necessary functions. Let's talk about writing the client part of the intelligent system. There are practically no analogues used in the JavaScript programming language, which in its essence is quite ambiguous. It has many pros and cons in use, has many program elements challenging to use, and requires practical knowledge to work with this programming language fully. We distinguish a pointer to the current object inside the function, closures, functional constructors, features of creating and writing asynchronous code. It requires knowledge of two different patterns for creating and processing asynchronous code at once.

JavaScript is also a functional, weakly typed programming language. Respectively, the value of all variables is determined dynamically in real-time, which is a feature of using the JavaScript programming language. On the one hand, it is very convenient and allows you to maximize the use and writing of code, provides many opportunities for processing, transmitting, and storing structured data. It also has many disadvantages, namely problems with data processing. It isn't easy to properly save and unpack data if it is unknown precisely what type of data was received for processing.

With the JavaScript programming language, an add-on is also actively used, namely the TypeScript programming language. It is very convenient and efficient to use. It is an OOP entirely programming language that provides the full functionality of the OOP basic principles, namely encapsulation, abstractions, imitation, virtual methods, interfaces, enumerations, and template data collections. They allow writing effective code in parallel with functional JavaScript. TypeScript is similar to the C# programming language. It effectively uses “reactive” code-writing methods based on asynchronously processed events and techniques, allowing asynchronous results. Using the JavaScript and TypeScript programming languages themselves does not allow writing fully functional code that will be fully structured and responsible for all templates. The Angular framework is quite popular but not as popular as its React and Vue counterparts.js. In this case, it is advisable to use it since only Angular provides complete functionality for writing highly structured and loosely coupled code, which will be easily expanded when adding or rewriting the program, adding new modules and components.

Angular uses TypeScript and is a fully OOP programming language, and it also uses basic modern mechanisms for writing and maintaining the codebase. For example, the entire intelligent system is created using the Single Page Application template (SPA). It builds dynamic programs that run on just one web page and uses asynchronous loading of all modules with display tools and using “reactive” operation of all data processing components: services, data stream handlers, data identification protection, and others. The implemented dependency allocation mechanism allows writing code based on the efficient use and substitution of program interfaces and their initialization inside the constructors of the selected service. It also allows writing code that is easy to redo or replace in the future, and it is highly efficient in its work. Component construction is another feature of the program under development. Each system element is a separate component, consisting of other parts; between members, data transfer is implemented in different directions, allowing the system to transfer data to both parent components and children. The decomposition of the developed intelligent system works, making it possible to consider each piece separately. If there are errors, quickly track and correct them without damaging other system components. It is impossible to write code for creating an intelligent system without using specific application software tools. Nowadays, many programs are designed to simplify writing program code. Most people avoid using regular text editors and use a particular IDE environment for specific programming languages.

We used an IDE environment developed by Microsoft – Visual Studio to write an intelligent system for socializing individuals by common interests. Visual Studio and the C# programming language are from the same developers, and the environment is the most convenient for using the C# programming language. The rider is the only analogue which is also a popular IDE for writing code in C#. It is cross-platform and available on almost all operating systems. Its primary disadvantage is that it is paid. It is also worth highlighting that Rider demands a significant load on the operating system, which means encountering some performance problems. Thus, it will be inconvenient to write code, open various files, assemble all available system components into one product, and other issues.

Visual Studio is available on all operating systems except Linux. Visual Studio also has a free version, which is entirely sufficient for most of the tasks set. This development environment is

convenient and easy to use and provides many different opportunities for writing code. It easily debugs a program simply using code execution breakpoints. There are many opportunities for writing tests and integrating other programs inside. The internal server built directly into the development environment performs all the necessary functions and provides permissions for running server applications, thereby facilitating the programmer's work.

A screenshot of the Visual Studio program and its main elements is shown in Fig. 22.

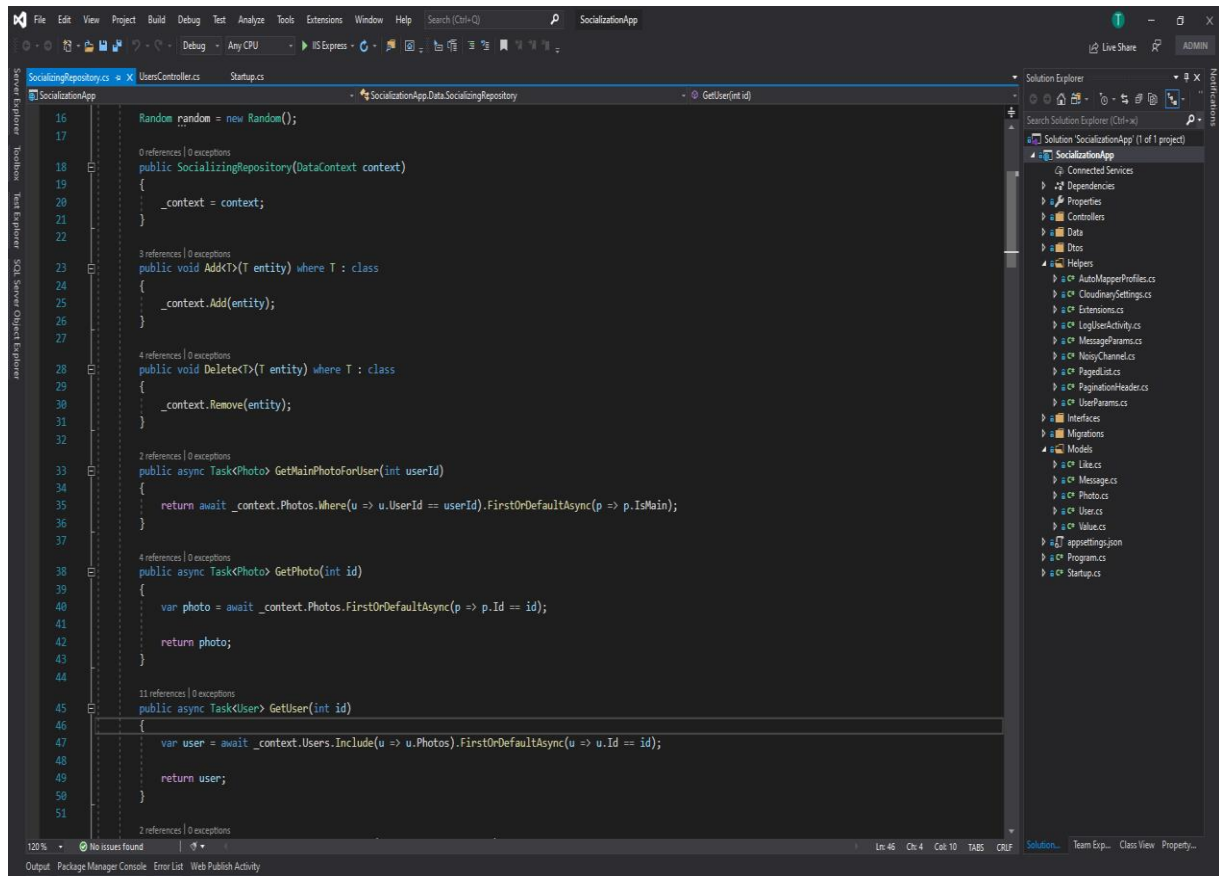
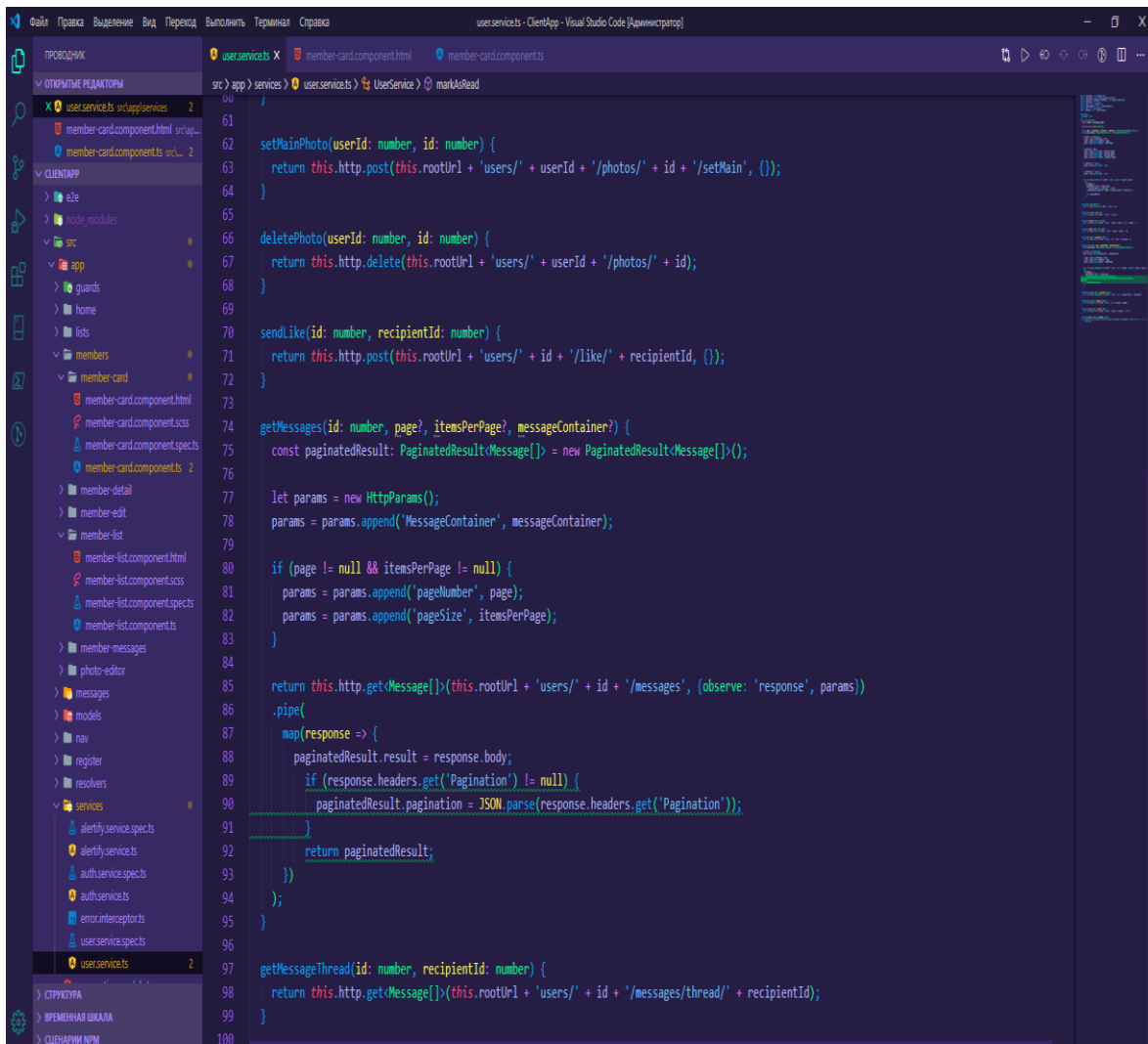


Figure 22: Visual Studio program operation

Unique environments are also used to write the client part of the program. In this case, there is already a lot to choose from: Visual Studio Code, Sublime Text, Brackets, and other application programs that are most often used for writing client software products. In this case, Visual Studio Code was chosen since Microsoft created this development environment, easily interacts with other company software products, and is very similar to other company products. Its main advantages are cross-platform and availability on all operating systems. It is convenient to develop an intelligent system everywhere, and no need to choose another one on which the environment is available. It is also completely free and provides all the features in total, which is convenient for all software developers. It is easy to operate, does not load the operating system, and rarely freezes. Development on it occurs quickly and without unnecessary pauses. Visual Studio Code is constantly updated, and all problems with using the product are fixed promptly by developers, which is convenient for users. Visual Studio Code is simple and easy to use, has a minimum of tabs, and can be easily and quickly understood by any programmer. Also, this development environment gives access to a fully automated build of software packages, which allows to save energy and focus on writing code. It is also possible to fix errors dynamically, using the page reload mechanism, which helps not stop the program execution but change specific properties on the go and reload the selected browser page to load the modified page components. Its disadvantages are inappropriate error handling, difficulty to understand exactly where the error occurred, search for problems inside the browser console, and ways of displaying the error. Still, in general, the environment is convenient to use.

A screenshot of the Visual Studio Code program is shown in Fig. 23.



**Figure 23:** Visual Studio Code program operation

It is also necessary to highlight databases essential for convenient and structured data storage among the software tools. In my case, they are MySQL, PostgreSQL, and Microsoft SQL Server.

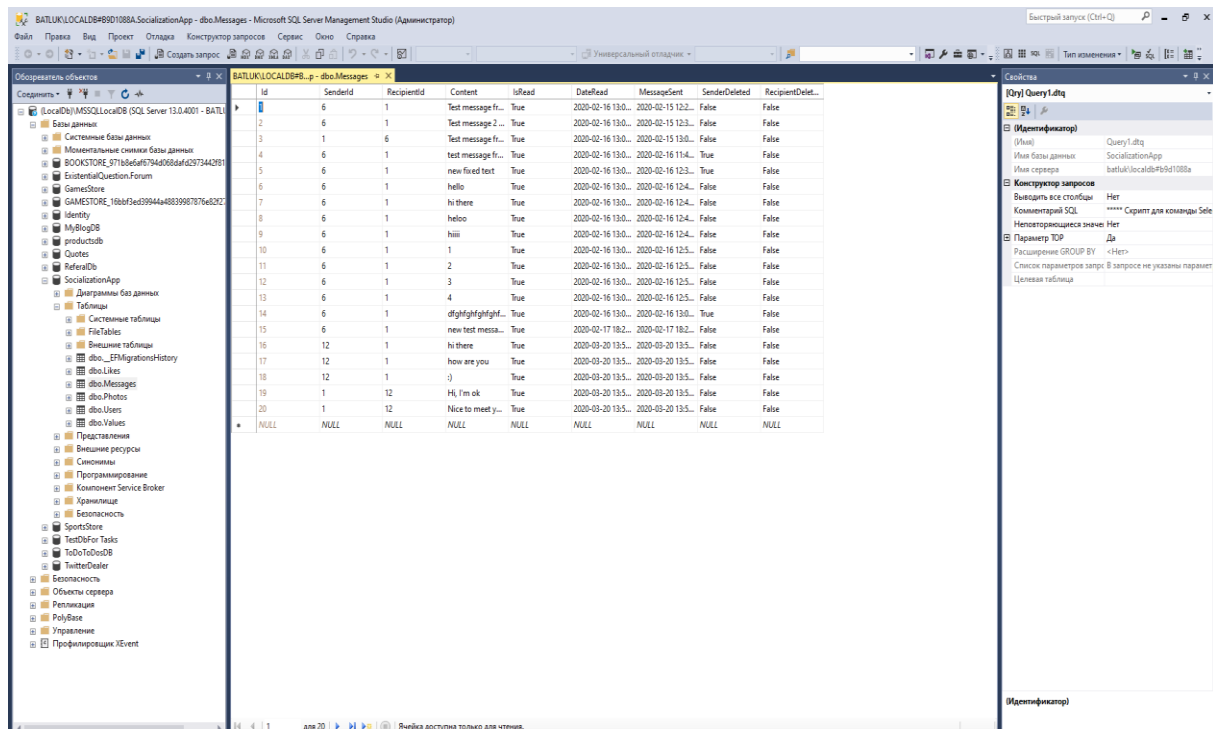
MySQL is one of the most popular DBMSs, which consists of a whole program package. The main advantages are free distribution, clear and straightforward documentation, and many functions for the DBMS user. The MySQL package is initially installed in standard repositories of the Linux operating system, supports a set of basic user interfaces, and can work with other databases. Among the disadvantages, it is worth highlighting the lack of performing some elementary functions available in other DBMSs. There is no support for OLAP, and there is no ability to work with XML files.

Among the advantages of PostgreSQL, it is worth highlighting scalability, which allows processing a large amount of information. This DBMS is entirely free, supports XML and JSON format. It also has several user interfaces and essential functions for managing and processing user data. The disadvantages include the lack of clear documentation, lack of initial configuration, and relatively low data processing speed compared to other DBMSs. Microsoft SQL Server is a popular DBMS created by Microsoft. Its advantages are ease of use, speed, and stability in operation; there are convenient tools for visualizing databases, even on mobile devices. It is also worth noting that SQL Server interacts conveniently and quickly with other Microsoft products. Among the disadvantages, it is worth highlighting poor optimization, which often loads the system to the maximum, poor implementation of file export and import services resulting in poor import or export of a file, or performing these actions with errors. We highlight all the advantages and disadvantages of existing DBMSs in more detail using a comparative Table 7:

**Table 7**  
Comparison of existing DBMSs

| Name                         | MySQL | PostgreSQL | Microsoft SQL Server |
|------------------------------|-------|------------|----------------------|
| Free distribution            | +     | +          | +                    |
| Documentation availability   | +     | -          | +                    |
| Basic functions              | +     | +          | +                    |
| XML support                  | -     | +          | +                    |
| OLAP support                 | -     | -          | +                    |
| JSON support                 | +     | +          | +                    |
| User interfaces              | +     | +          | +                    |
| Basic availability           | +     | -          | -                    |
| Working with other databases | +     | -          | -                    |
| Easy to use                  | -     | -          | +                    |
| Fast execution               | +     | -          | +                    |

After analyzing all the application servers, it was decided to choose Microsoft SQL Server. It is easy to configure and agrees with other Microsoft products. SQL Server Management Studio is selected as the software development environment for the Microsoft SQL Server. This software environment is visually convenient, and almost all actions with the database can be performed using the capabilities of the environment itself and using scripts. Also, it is very advanced in terms of configuration, making it easy to develop an intelligent system. It is straightforward to set all the requirements and issue access rights in the environment since it is essential to create all database accesses. One of the disadvantages of this environment is its absence on all popular operating systems, except for Windows, which is inconvenient for many users. SQL Server Management Studio is also heavy in terms of the system load. With a complex workload, the environment starts to run slower and slows down the person currently using it. The workload of the environment regarding options and features is a significant benefit for experienced users; otherwise, it is difficult to understand for beginners – they often need to check its functions on the Internet. A screenshot of the SSMS program is shown in Fig. 24.

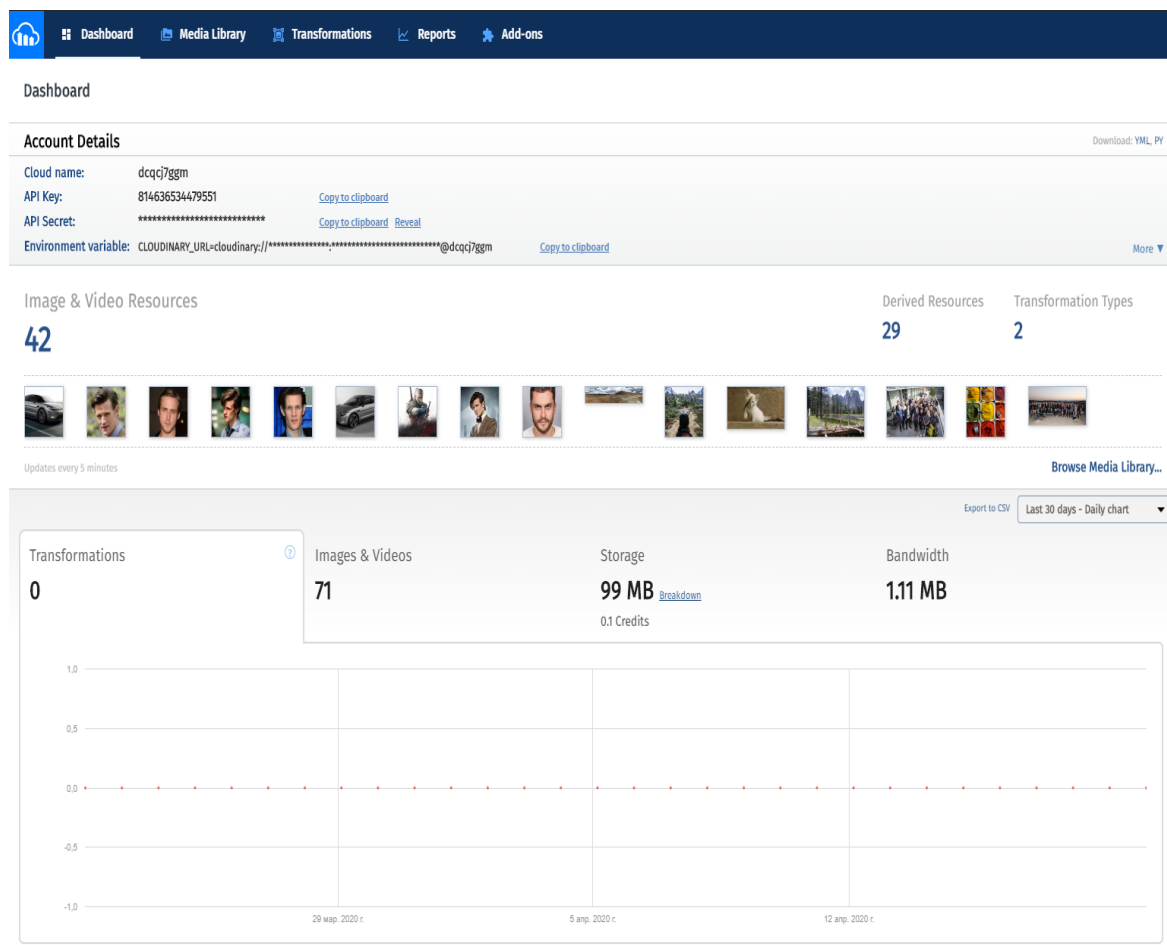


**Figure 24:** SSMS program operation



When working with a database, it is often difficult and inconvenient to work directly using queries, which can cause many problems. In this case, an ORM is used. It means that instead of working directly with data, an additional layer of abstractions is added. Data are presented as objects. Accordingly, working with data is carried out similarly to working with collections of objects in C#, which is very convenient in writing an intelligent system for socializing individuals by common interests.

An ORM called Entity Framework Core is chosen. This choice was made because it is easy to use, has clear documentation and a support system. This framework is convenient and intuitive to use, has many working methods, and can use asynchronous analogues of all forms of working with data. Entity Framework Core has one drawback – it is slow in data processing since migrations are created, additional resources are spent on data processing, and extra time is spent generating new data. Sometimes a problem of optimizing specific queries occurs inside the created migrations. It is also worth noting the use of a Cloud server for storing data that takes up a lot of space and is not suitable for storing in a database. The Clouinary service is chosen among the many options; it has a free subscription, including all initial requirements. A screenshot of the cloud environment is shown in Fig. 25.



**Figure 25:** Clouinary work

There is a library specifically for the C# programming language, convenient to use after studying the documentation. Also, this Cloud server is quite fast in terms of processing requests and returning generated data.

The choice of tools for solving the problem was discussed. The main reasons for choosing and the advantages of particular programming languages, tools for creating program code and data processing were indicated. The main technical characteristics of the software tools selected to develop the current intelligent system for the socialization of individuals were also demonstrated and described.

## 5. Results

### 5.1. Description of the created software tool

If you describe the intelligent system for the socialization of individuals by common interests, you should start with the implemented mechanisms for registering and authorizing users of the system. User registration and authorization are implemented using the Identity mechanism, which provides an API for creating a convenient system for working with various intelligent system profiles. The Register method is used to register users, which accordingly performs creating a user profile. It accepts two necessary parameters for input: a variable of the User type, which contains all the properties entered by the user for saving in the database, and a variable password – the password entered by the user. Encryption is used to protect the user's password. Two variables of the byte array type are used to implement encryption: passwordHash and passwordSalt.

We chose HMACSHA512, a modern cryptographic mechanism for checking the integrity of the information that uses hash functions and guarantees data integrity. During the algorithm's execution, a new object is created, its key is assigned to the passwordSalt variable, and the transmitted password hash is calculated in passwordHash. PasswordSalt is a particular string containing data that is transmitted simultaneously with the input data array and an image for calculating the hash.

After returning the created passwordSalt and passwordHash from the method, we write them to the User object passed earlier to the technique and asynchronously writes a new user to the database. It asynchronously updates the database and saves changes, and the created user returns at the end. Also, before making a user, its presence is already checked using the UserExists method.

It is relatively easy to authorize a user at the server level, and it has specific features in the client part of the intelligent system. On the server-side, the user's username and password are passed to the Login method. It checks whether a user who has such a username and password exists in the database. If such a user does not exist, then null is returned. If the user exists, then the password hashes are compared using the VerifyPasswordHash method. The current user password is passed, and passwordHash and passwordSalt are created in the user method taken from the database. If the hashes match, the user object is returned from the way.

After that, authorization is performed using the Json Web Token (JWT). It is a relatively new technology that allows you to conveniently log in to the user by storing part of the token in the browser's database. First, the required permissions are determined for the user according to the access level using the properties of ClaimTypes.NameIdentifier and ClaimTypes.Name. Next, a token is extracted from the protected file, the encryption key, and a symmetric access key is encrypted. Then, the primary user properties are displayed by creating a separate signature. The next step is to create a custom token handle – this is an object that contains all the permissions for the current user, the validity period and validity of the current token for the user, and the user's properties with the signature. After that, an object of the JwtSecurityTokenHandler type is created, responsible for generating the token itself. Next, we create a token using the data that is in the made handle. The last step is to return an anonymous object from the method, which contains the created token and an object with all the properties of the current user. The client receives a request from the server and uses the AuthGuard directive to check the validity of a part of the token intended for storage inside the browser's local database. An asynchronous process of saving the token occurs. All data is stored in tables in a relational database, and all available information is linked to each other using relationships. A diagram of the current database is shown in Fig. 26. First of all, the key table of the database should be described – Users, which represents the current user of the system. It has the following main fields: Id-unique – user ID, UserName – user login, PasswordHash – hash of the user's password, PasswordSalt – user hash image, Gender – user's gender, Date Of Birth – date of birth, KnownAs – user's nickname inside the system, Created – registration date, LastActive – date of last user visits, Introduction – introductory information about the user, LookingFor – basic information about what exactly the user needs and what they are looking for, Interests – description of the system user's interests, City – city of residence, Country – country of residence, SimilarPercent – temporary parameter of user similarity. Also, as you can see from the database diagram, the Users table is linked to other tables: Photos, Likes, and Messages using a one-to-many relationship; respectively, Users is the central table.

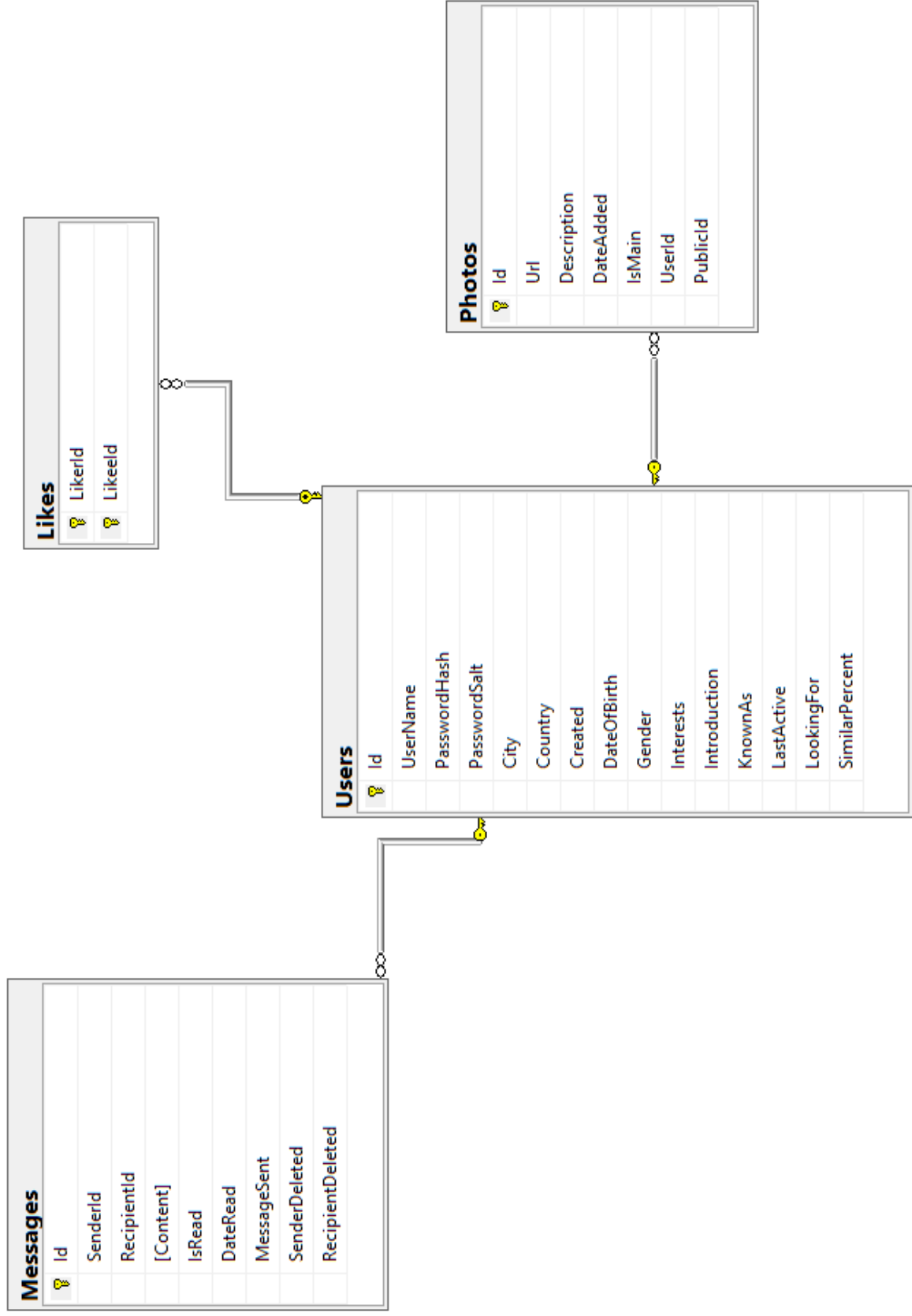
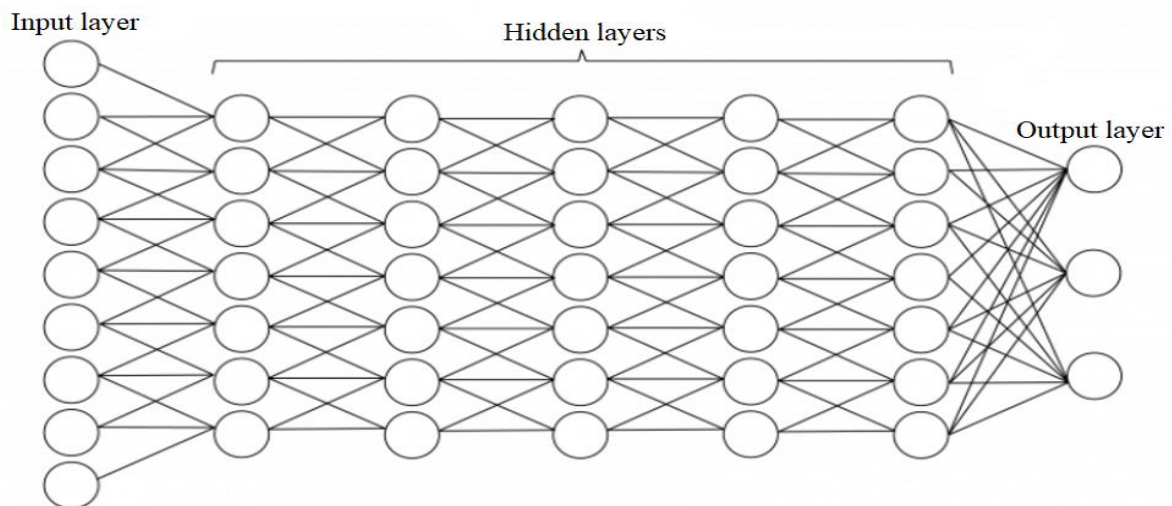


Figure 26: Intelligent system database diagram

The Photos table contains all the main properties of users' photos in the intelligent system: Id – a unique identifier, Url – link to the image, Description – Description, DateAdded – date added, IsMain – a boolean field that defines the user's avatar, PublicId – ID of finding the image in the cloud storage service and UserId – foreign key, which means that the image belongs to one of the users. Next, the Likes table contains the Ids of users who put likes, namely LikerId – a unique identifier of the user who likes and LikeeId - a unique identifier of the user who was selected in the system.

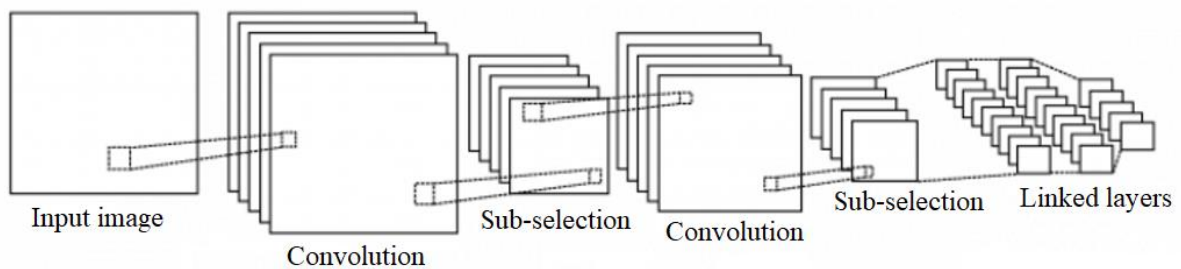
The Messages table contains all the basic properties of messages sent by users of the intelligent system: Id – unique message ID, SenderId – unique sender ID, RecipientId – unique recipient id, Content – message content, IsRead – marked or read messages, DateRead – read date, MessageSent – sending date, SenderDeleted – deleted by the sender, RecipientDeleted – deleted by the recipient.

The implementation of a neural network [54-55] that searches for faces in users' photos [56-60] and compares them with those already available in the current database is one of the most critical implemented algorithms in the intelligent system for the socialization of individuals by common interests. For this task, it was decided to use deep neural networks [54-55]. Its main feature is that in addition to the input and output layers, they also consist of a certain number of hidden layers following the requirements of the intelligent system implementation. An approximate view of the deep neural network architecture is shown in Fig. 27. If we talk about a deep neural network for face recognition, first of all, the concept of dividing layers into two completely separate sub-selections is chosen. They are divided into convolutional layers and discretizing layers [61-63]. Due to this approach, each neuron corresponds to a distinct point in a fixed image and connects with the receptive field, that is, with the area of the image fed to the input.



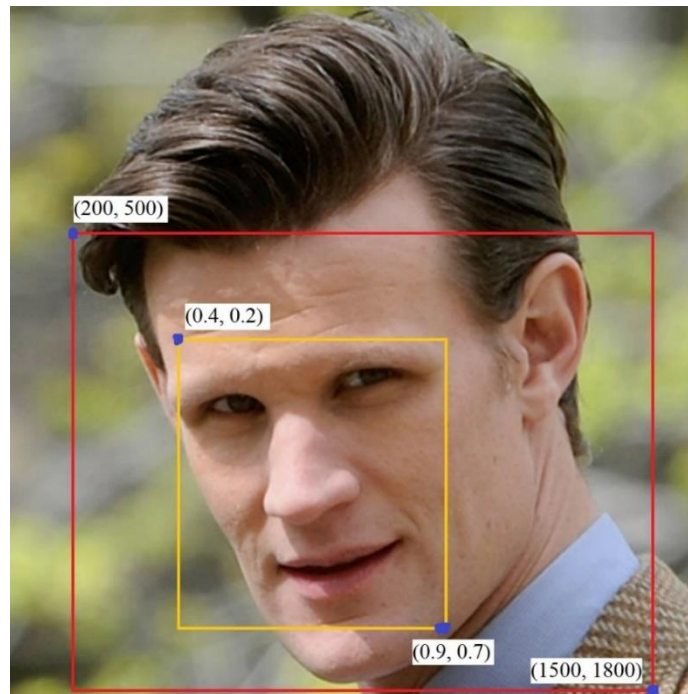
**Figure 27:** Neural network with hidden layers

Each location of the selected layer is distributed to a certain number of neurons. Each is unique since it has a unique set of input weights formed using neurons in the previous rectangular layer before the current processing layer [54-55]. If we draw the entire structure described visually, the convolutional neural network will look like in Fig. 28.



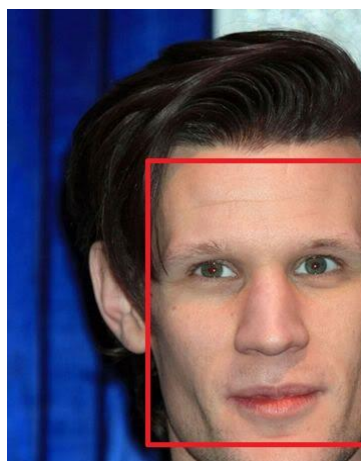
**Figure 28:** Structure of a convolutional neural network

The following procedure should be followed to implement a convolutional network: upload a photo or several photos to the program, save the uploaded images using asynchronous processing in the database, and then start processing. A convolutional neural network consists of five main layers: the input layer, the output layer, and three internal hidden layers (the P-Net layer, the R-Net layer, and the O-net layer). Each layer processes the uploaded image sequentially. First, the P-Net layer performs processing, highlighting two square frames in the photo, as shown in Fig. 29.



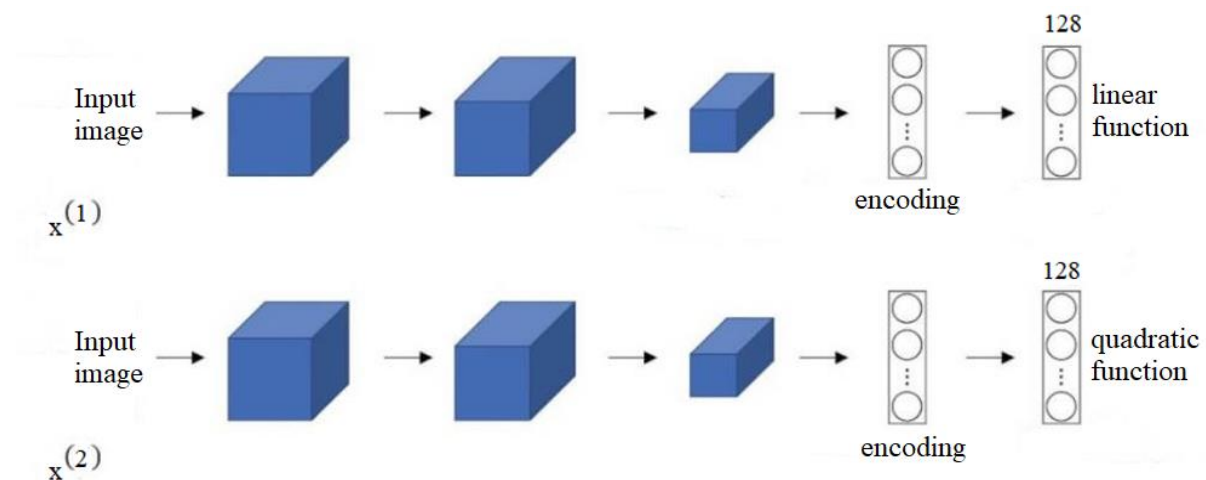
**Figure 29:** P-Net layer

In this image, the red box shows a 24x24 core that has changed in size to the original image. Calculated core width and height:  $1500 - 200 = 300$ ,  $1800 - 500 = 300$ . The width and height obtained here are the core's width and height up to its original size. Then the coordinate frames are multiplied by 300:  $0,4 \times 300 = 120$ ,  $0,2 \times 300 = 60$ ,  $0,9 \times 300 = 270$ ,  $0,7 \times 300 = 210$ . The upper-left coordinate of the kernel is added to get the coordinates of the boundary field:  $(200 + 120, 500 + 60)$  and  $(200 + 270, 500 + 210)$  or  $(320, 560)$  and  $(470, 710)$ . The received data is saved, and then the R-Net layer starts working. For each bounding box, an array of the same size is created, and the pixel values are copied to the new array. Sometimes the image may contain only part of the face peeking out of the frame, as shown in Figure 4.5. In this case, the R-Net layer may return a bounding box partially out of the frame.



**Figure 30:** R-Net layer

If a bounding box does not go out of bounds, an image part is copied to the new array, and the rest is filled with 0. This process of filling arrays with 0 is called padding. After the bounding box arrays are plotted, their size changes to 24 x 24 pixels and is normalized to values from -1 to 1; currently, the pixel values range from 0 to 255. Each pixel value is subtracted by half of 255 (127.5) and divided by 127.5, after which the value is stored between -1 and 1. After standardizing the coordinates, the bounding boxes are rearranged to the square to be passed to O-Net. The received data is saved, and then the O-Net layer starts working. The O-Net results are slightly different from the P-Net and R-Net results. O-Net provides three outputs: the bounding box coordinates, the coordinates of 5 face landmarks, and the confidence level of each field. By getting rid of boxes with a lower level of trust, both coordinate frames and face reference coordinates are standardized. The final step is to pack all the information into a dictionary with three keys: field, confidence, and key points. The “field” contains the coordinates of the bounding box, the “confidence” contains the network confidence level for each window, and the “key points” include the coordinates of each face reference point (eyes, nose, and mouth endpoints) [64-81]. The next step is to use the Siamese neural network, responsible for searching for similar faces in existing photos in the database. The structure of the Siamese neural network is shown in Fig. 31.



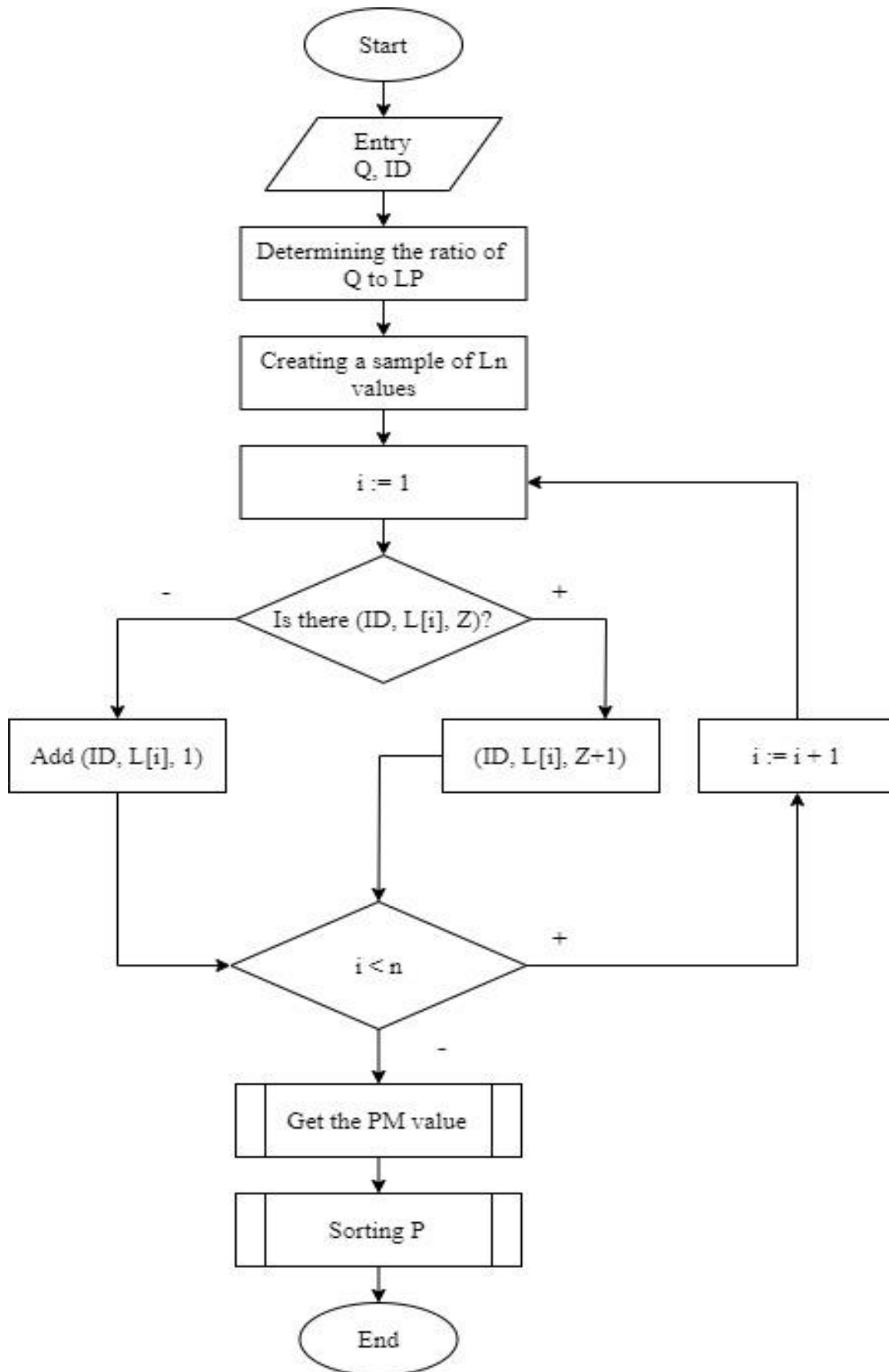
**Figure 31:** Structure of the Siamese neural network

The Siamese neural network consists of two identical layers, each with the same weights. Each of the layers takes the same image as input data. At the output of each layer, linear and quadratic functions compare the image with the existing one inside the database and form a particular coefficient. The last step is to subtract the difference between the coefficients and determine the presence of the picture.

The primary function of the intelligent system for the socialization of individuals by common interests is to search for relevant users, so the main task was to write an optimized algorithm that automates the process of user's socialization as much as possible [82-85]. In this case, a special algorithm is created based on algorithms such as the Levenshtein algorithm, the sample expansion algorithm, the n-gram algorithm, and the Noisy Channel models [82-91]. First of all, we select the input data that should be used to generate the final sample for the user. This data is the system user parameters, namely the user's description, the user's interests, and the field that describes who the user exactly searches [82-85]. These parts will be used for socialization. From the very beginning, an extended sample is formed based on the disbanding of the presented data array into individual elements using the sample expansion algorithm [86-91]. This algorithm is used most often in all word comparison programs. Its peculiarity is that it discards all found results without matches and postpones them for fuzzy search algorithms, and it searches for the values of the obtained elements. For each word received from the initial sample, a set of values is constructed, based on which the search for other terms is implemented. This algorithm essentially divides the piece into two sub-samples, one of which will be processed by the main fuzzy search algorithms, and the Noisy Channel model will process the other.

The algorithm was further modified to generate intermediate variants of text processing using special rules with the rejection of word endings to avoid incorrect division into subsamples due to an error in word structuring from the beginning of text analysis. This modification allows you to significantly

optimize the algorithm's operation since you can avoid using complete data collections for storing text and dynamically feed data in separate elements. A flowchart of the algorithm is shown in Fig. 32.



**Figure 32:** Sample expansion algorithm

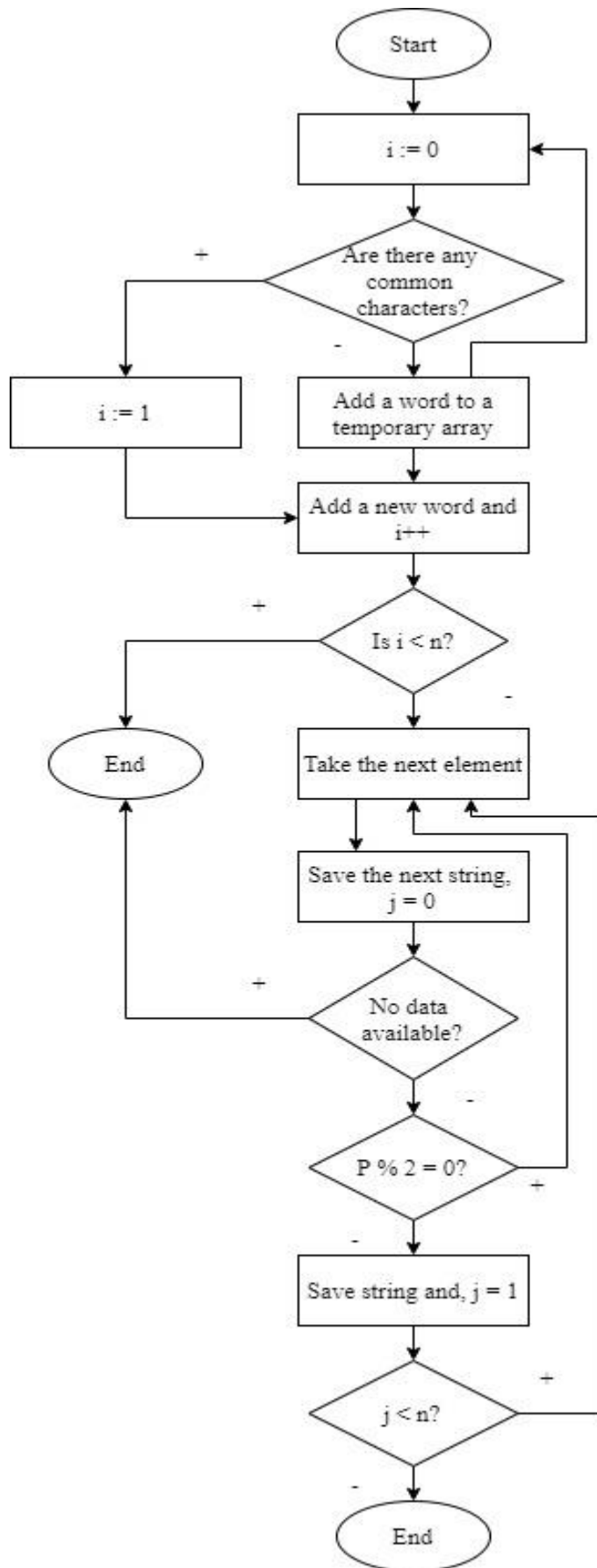


Figure 33: N-Gram algorithm

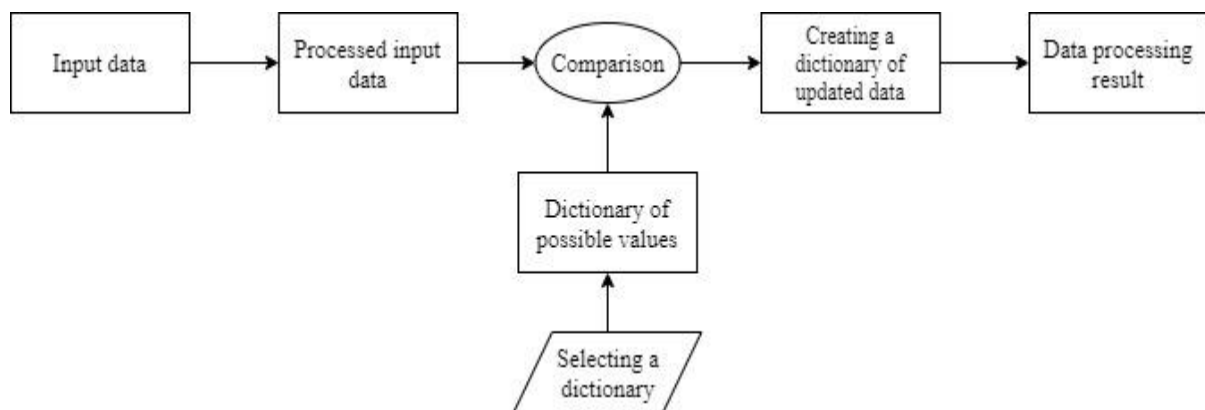


After dividing the data into two subsamples, the first subsample is taken, and the N-Gram algorithm is used to process it. The algorithm itself is pretty old and has been used for a long time. It is simple to implement, easily modified according to specific requirements, and works relatively quickly. Thus, it is the fastest fuzzy search algorithm among those used in the intelligent system to socialize individuals with shared interests. If you describe the algorithm's operation, it takes the generated data array and compares it element-by-element with the compared data array. The comparison itself is carried out using a straightforward formula. If word one coincides with word two, taking into account some errors, then there is a high chance that they will have a standard string of length N. A flowchart of the algorithm is shown in Fig. 33.

While it is indexed, the desired word is divided into such N-grams that the average probability of similarity of a word match is more than 50% and falls into a separate list of words. Most often, trigrams are used as strings that consist of three letters. The algorithm has certain disadvantages. They may occur when creating all the trigrams of lines. Many words can fall out of the sample, even if they fit the search parameters of the word since there may be a word that does not match the division and will be mistakenly not selected by trigrams before that sample expansion algorithm was used to minimize the possibility of having similar words in the first subsample.

It is believed that most of these words fell into the second sample, so the Noisy Channel model is used with it. It is designed to process all the words that were most likely distorted in a certain way. It treats them as an additional dictionary of possible meanings, which will be another place to store and compare words from the second subsample, so a collection of possible distorted words is developed.

The Noisy Channel model is prolonged to use due to the constant formation of a middle dictionary during text processing. To optimize it, we created a selection of most likely distorted words not to waste extra resources. A flowchart of the Noisy Channel model is shown in Fig. 34.



**Figure 34:** Noisy Channel Model

The operation of the Noisy Channel model is shown on the example of the word “actress.” The idea is to replace similar words using certain word processing operations. If there are no false letters, the correct term is selected, as shown in Table 8.

**Table 8**  
Example of how the Noisy Channel model works

| Error         | Possible fix   | Correct letter | False letter | Type            |
|---------------|----------------|----------------|--------------|-----------------|
| <b>acress</b> | <b>actress</b> | <b>t</b>       | <b>No</b>    | <b>Deleting</b> |
| acress        | crress         | -              | a            | Insert          |
| acress        | caress         | ca             | ac           | Transposition   |
| acress        | access         | c              | r            | Subtraction     |
| acress        | across         | o              | e            | Subtraction     |
| acress        | acres          | -              | s            | Insert          |
| acress        | acres          | -              | s            | Insert          |

The internal algorithm for processing fully formed word samples is the Levenshtein algorithm. The main point of the algorithm is to determine the distance between several sequences of characters. The algorithm at the output determines a particular value, namely the number of necessary substitutions for one word to match the compared word in case of differences. It forms a percentage that shows the chance of word combinations and determines whether two similar words got into the sample. The algorithm is optimized using a finite automaton, so all intermediate data is deleted from memory with a change in iteration. First, there is a formation of two subsamples of words out of the users' data input. Each of the two subsamples is additionally worked out to increase the accuracy of the word search. Then, using the Levenshtein algorithm, there is a direct comparison of word samples of all system users, and each word is compared with the corresponding parameters of the user under study. A flowchart of the algorithm is shown in Fig. 35.

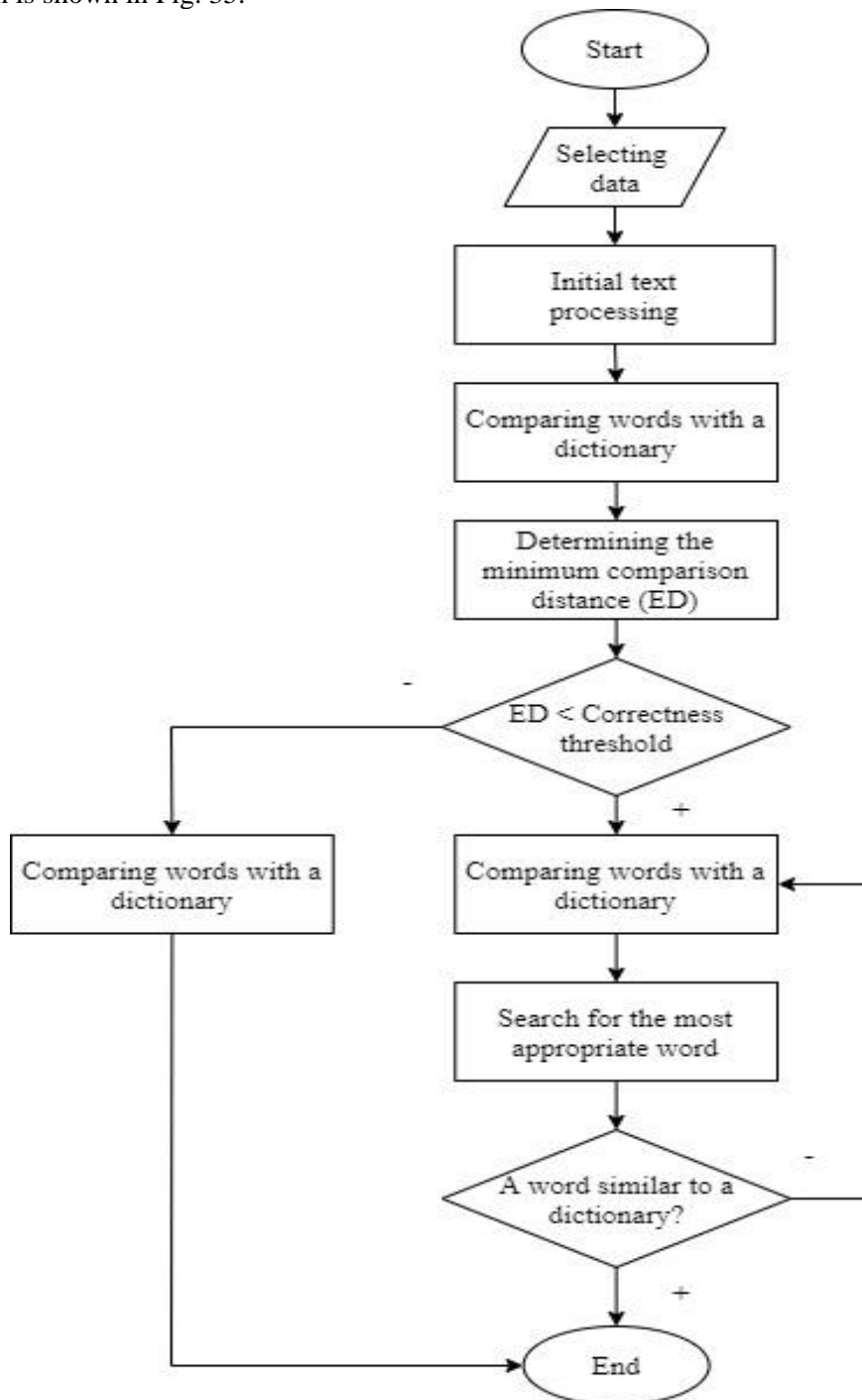


Figure 35: Levenshtein Algorithm

During the comparison, points are generated for each match: three points for matching interests, two points for the search parameter, and one point for general information. Before starting the Levenshtein algorithm, the maximum score of the current user is developed since all unnecessary words have already been thrown out, and we can assume that the score is formed based on keywords. The Levenshtein algorithm work is shown while comparing the words “elephant” and “relevant” in Table 9. When each word matches, users are awarded points, which are summed up with each iteration. The last step is to compare the total score of the current user and the searched one, based on which user similarity is formed. Having received a ready-made sample of users, we can interact with them. The intelligent system for the socialization of individuals with shared interests provides all the necessary functions that are intuitive for the average user. So, the GetUsers method is designed to display the generated list of system users. First, the current user is searched so that his profile is not thrown into the sample, and then his ID is excluded from the available selection.

**Table 9**  
Example of how the Levenshtein algorithm works

|   |          | E        | L        | E        | P        | H        | A        | N        | T        |
|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|   | <b>0</b> | 1        | 2        | 3        | 4        | 5        | 6        | 7        | 8        |
| R | <b>1</b> | 1        | 2        | 3        | 4        | 5        | 6        | 7        | 8        |
| E | 2        | <b>1</b> | 2        | 2        | 3        | 4        | 5        | 6        | 7        |
| L | 3        | 2        | <b>1</b> | 2        | 3        | 4        | 5        | 6        | 7        |
| E | 4        | 3        | 2        | <b>1</b> | <b>2</b> | 3        | 4        | 5        | 6        |
| V | 5        | 4        | 3        | 2        | 2        | <b>3</b> | 4        | 5        | 6        |
| A | 6        | 5        | 4        | 3        | 3        | 3        | <b>3</b> | 4        | 5        |
| N | 7        | 6        | 5        | 4        | 4        | 4        | 4        | <b>3</b> | 4        |
| T | 8        | 7        | 6        | 5        | 5        | 5        | 5        | 4        | <b>3</b> |

The directive Response.AddPagination generates the server's response to the request from a sample of users. It passes the current page to display users, the page size, the number of elements, and the total number of pages. The method returns the generated user sample following the original parameter, which was used to create the piece with all the generated page data. It is followed by the GetUser method, which returns a specific user to view the user's profile. The technique takes a unique user ID and returns the found user in the current database or returns a response that an error occurred and the user with this ID does not exist. The UpdateUser method follows it and takes as parameters both the user's ID whose data you want to update and the object which contains all the updated data sent from the application client. First of all, it checks whether the current user is available in the database and logged in. Then the database is accessed, and information about the user is updated asynchronously. If there is a storage error in the database, an exception is thrown. The code and cause of the error are also logged inside the database and stored in the appropriate list. The next step is to apply the LikeUser method, designed to mark the user's preferences. First, it checks the system user authorization, and if the current user is not registered, then an error message is thrown, which does not allow performing the same action. Otherwise, if there are no errors, a new Like object is created, added to the database, and a request is sent to the user registered. The next step is to apply the CreateMessage method for sending messages. It has two main parameters: the sender ID and the object of the message itself. The sender is searched in the database, then if the search is successful, the sender ID is saved, and the recipient of the message is searched using the database ID. If the search is successful, the message itself is generated. All message properties that need to be saved are written to a particular variable that is storing data. First, the message is stored asynchronously in the database. Then an anonymous object is generated and sent to the users using the routing mechanism.

## 5.2. User manual

A unique user manual is created to understand how to use the intelligent system to socialise individuals with shared interests.

### 5.2.1. Introduction

*Goal.* The intelligent system of the socialization of individuals is designed to demonstrate the principles of interaction of system users. For this purpose, a neural network is used to identify users and text analysis algorithms to select suitable users.

*Abstract.* The intelligent system is designed for socializing users. The application task is to register users in the system, enter data, and upload their photos. The aim is to socialize users of the system. The input data is user information and user photos. The output data generated by the system is a selection of users sorted in descending order of the percentage of user similarity. Also, the program's features include the option to like the selected user, have private correspondence within the system, update all personal data, and view read and sent messages.

### 5.2.2. General information about the program

*Designation and name of the program.* Full name: Intelligent system for the socialization of individuals with shared interests based on SEO technologies and machine learning methods. Abbreviated name: Intelligent system for the socialization of individuals with shared interests.

*Programming languages.* C#, JavaScript, TypeScript, ASP .NET Core MVC, Angular.

*Program assignment.* Socialization of users by forming a percentage of user similarity based on the specified information.

*Program features.* The intelligent system identifies the user's face in the photo and puts a specific one as the main. It also performs user socialisation, forming a list of suitable users, selecting a user, putting a like, and carrying out private correspondence inside the system.

### 5.2.3. Classes of tasks to be solved

*Our objectives* are: registering and authorizing users inside the system, saving user information, saving user photos, analyzing user photos and searching for faces in photos, changing the primary image, forming a list of suitable users and forming a percentage ratio, saving data about selected users, keeping private messages, saving pictures in a cloud service, dynamically loading data.

*Methods for solving problems.* Identity tools and JWT tokens are used for registration and authorization, and information is saved inside the database using the Entity Framework Core Orm functionality. The Cloudary service is used to save photos, convolutional and Siamese neural networks are used to analyze images, internal functions of the system are used to change the cover photo, and a list of suitable users and forming a percentage ratio is performed using algorithms such as the Levenshtein algorithm, the sample expansion algorithm, the N-gram algorithm and the Noisy Channel model. Session retention methods, methods for storing private messages, and storing information of each message separately are used for user interaction. The task of dynamically loading data is performed using asynchronous methods and “reactive” programming methods.

*Functions.* The following procedures are used: Register – for user registration, Login –for user authorization, GetUsers – for generating a list of users and displaying the generated list, GetUser – for displaying complete information about the selected user, UpdateUser – for updating user information, LikeUser – for marking a preference for the selected user, GetPhoto – for choosing the main photo, AddPhotoForUser –for searching for a face and adding a photo, SetMainPhoto – for setting the main photo of the user, DeletePhoto – user photo deletion function, GetMessage – for selecting a single message from the list, GetMessagesForUser – for displaying sent, received, or unread messages, GetMessageThread – for showing all messages from the dialogue between users, CreateMessage –for creating and saving the message, DeleteMessage – for deleting the message, MarkMessageAsRead – for marking the message as read.

#### 5.2.4. Description of the main characteristics and features of the program

*Time.* The program optimizes all tasks performed since there are no page reloads and all requests are asynchronous. Photo analysis by neural networks occurs asynchronously using “reactive” programming. Text analysis algorithms are improved for processing large amounts of data. The system itself advances user socialization, as it generates the percentage of user’s similarities and allows choosing the best one.

*Operating mode.* The system is created as a web application. Accordingly, it works around the clock and starts working when the first user requests inside the system. The JWT token is used for the first request and registration/authorization of the user. A session is created for 24 hours, all intermediate data uploaded or selected are saved and cleared at the end of the session.

*Monitoring tools.* There are two types of errors in the system: data loading errors and page loading errors. Almost all system functions return either an error message or emergency message in the event of an error without damaging the entire system. If the page loads incorrectly, the user is redirected to a particular web page with an error message.

*Program limitations.* The main rules of the program are: it cannot guarantee the reality of the system user based on the photo, accurately generate the percentage of users, be used by an unauthorized user, process data offline.

#### 5.2.5. Functional restrictions on the application

*Conditions required for the program performance.* The main limitations are: a registered and authorized user can only use the program to create a profile. The system user must have their photo set as the main one with all the data filled in. The user should have access to the Internet.

*Technical and software tools.* The intelligent system is created in the form of a web application so, the user of the system must have a computer or phone with internet access and a browser.

*Parameters of peripheral devices.* Peripherals should include a keyboard and mouse for entering data and a monitor for displaying data.

*Software requirements.* Among the software tools, you only need a browser, preferably the latest version with Internet access and the function to use cookies. The user should install any working operating system and drivers for using the keyboard, mouse, and monitor.

*Organizational, technical, and technological requirements and conditions.* For the system to work correctly, you must have constant access to the internet, cookies should be enabled in the browser, and data input and output devices should function properly.

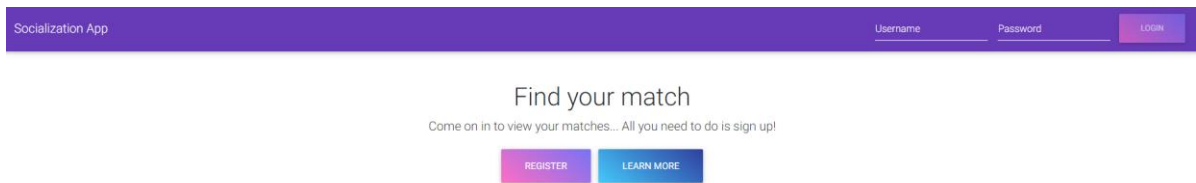
Under the standard regulations, a user manual is written. It is divided into two sections: an introduction and the primary information about the program. The introduction section indicates general information for the user, and the central part presents information about the program, its purpose, functions, and programming languages in it is developed.

The classes of tasks solved are described. They include the program's main functions, how exactly the program performs them, and the tools to solve all the necessary procedures are indicated. The main characteristics and features of the program are demonstrated: the time characteristics, operating modes, restrictions within the program, and the means of monitoring and self-restoring.

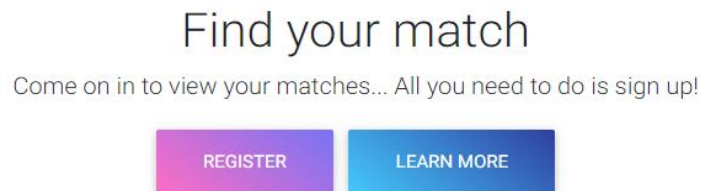
Information on the functional limitations of the program application is provided. It covers the optimal conditions for the correct execution of the program, the software and hardware requirements, and a list of peripherals and application software tools. In addition, the leading software requirements and the conditions under which the program can fully perform all the necessary functions are described.

### 6. Discussion

The control example shows the main functions and work of the created intelligent system. Fig. 36 shows the main screen of the program. Fig. 37 shows the user registration form. Fig. 38 shows the user registration form. Fig. 39 shows the user's authorization, login, and password entry, and Figure 4.15 shows a message upon successful authorization.



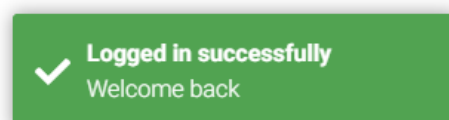
**Figure 36:** Main screen of the program



**Figure 37:** Buttons of the main program screen

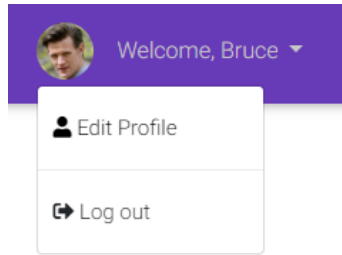
**Figure 38:** User Registration Form

**Figure 39:** User authorization

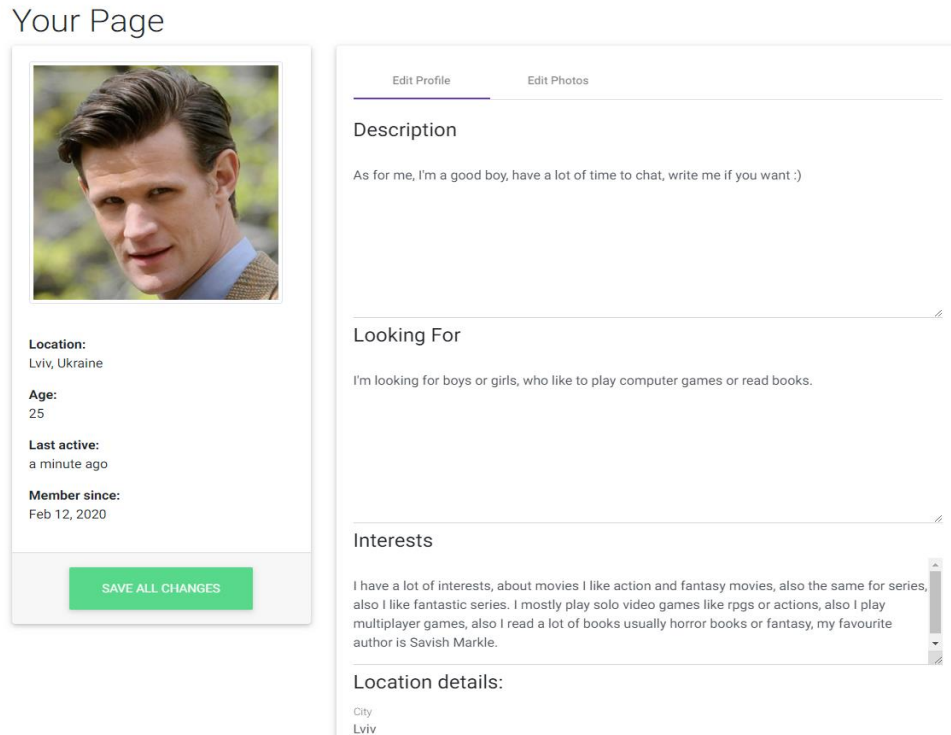


**Figure 40:** Message upon successful authorization

Fig. 41 shows the user profile settings, and Fig. 42 shows the completed user profile.

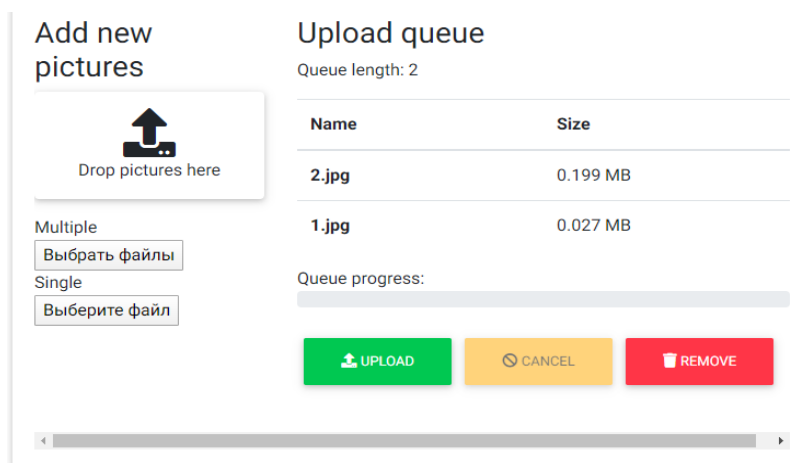


**Figure 41: User Profile Settings**



**Figure 42: Completed user profile**

Fig. 43 shows the process of uploading photos into the system. We can upload one or more pictures simultaneously by dragging them manually or using file explorer.



**Figure 43: Uploading photos**

Fig. 44 shows uploaded photos of the user. Users can delete all images except the cover photo, and neural networks also process all images. The image with no face on it is not available for displaying the user's cover photo. Fig. 45 shows the generated list of users using text processing algorithms and sorted in descending order by the percentage of user similarity.

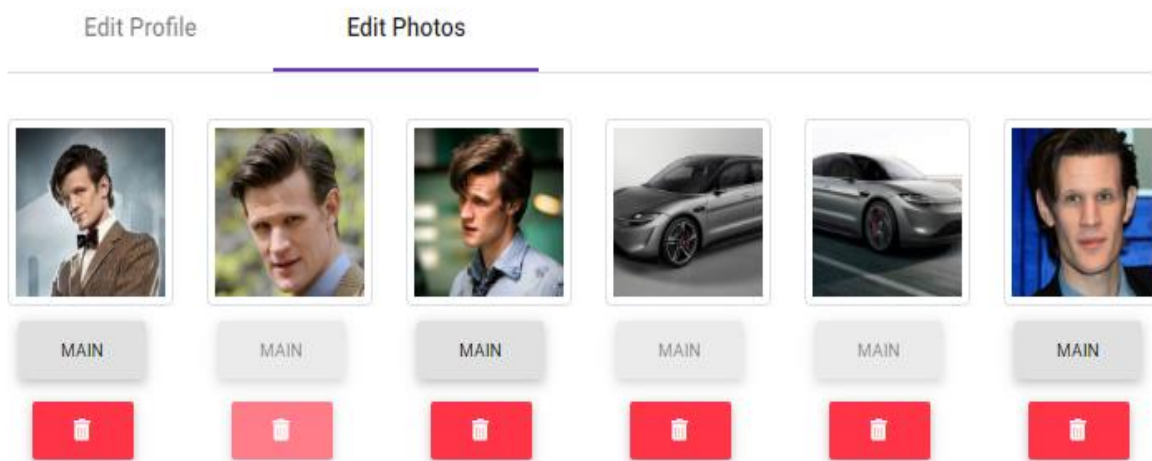


Figure 44: Uploaded user photos

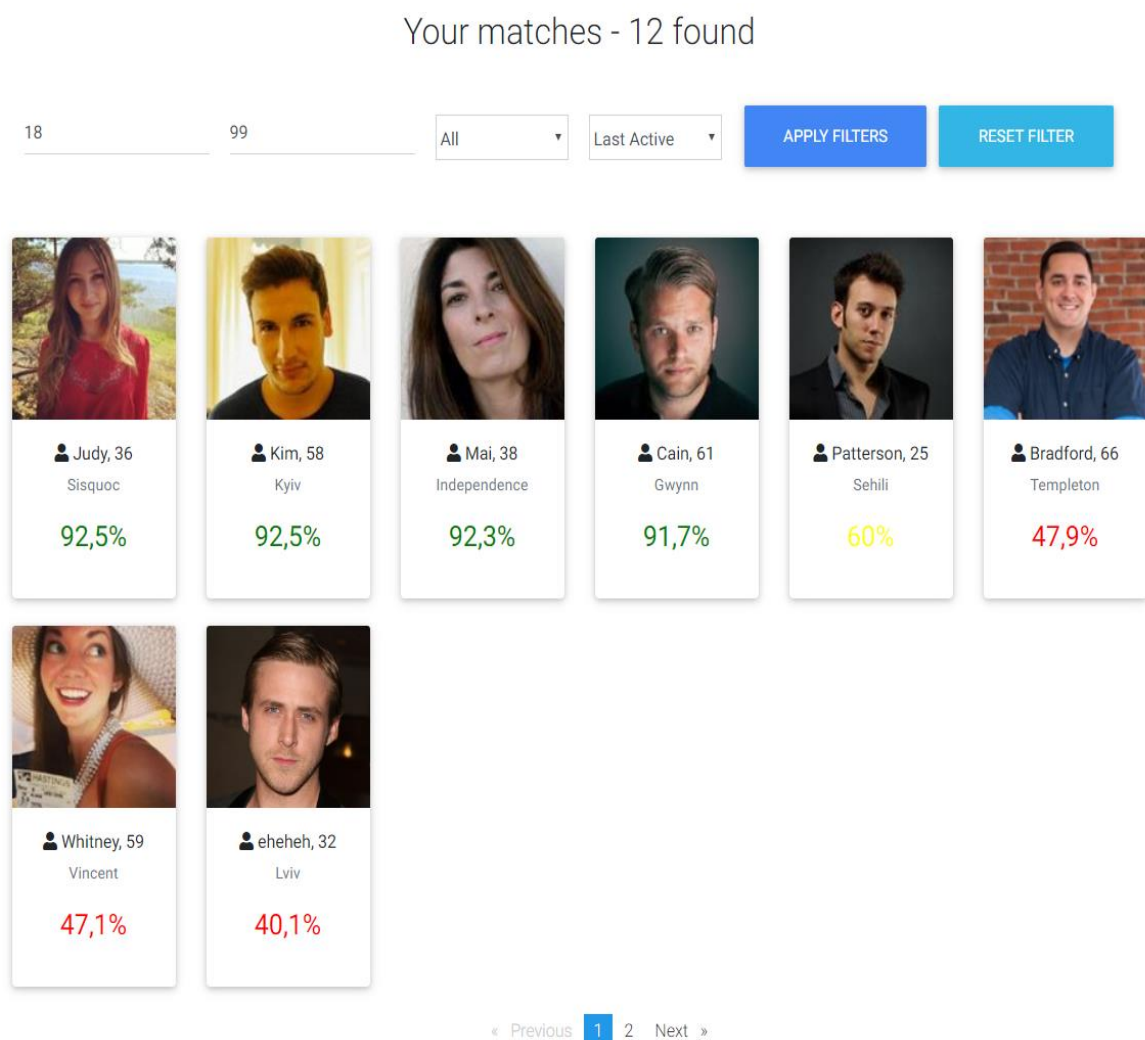


Figure 45: Generated list of users

Fig. 46 shows the use of filters to search in an already generated list. Fig. 47 shows the user profile selection, viewing the user profile and writing a private message. Fig. 48-49 24 show a tab with information about users' preferences who select and are selected.



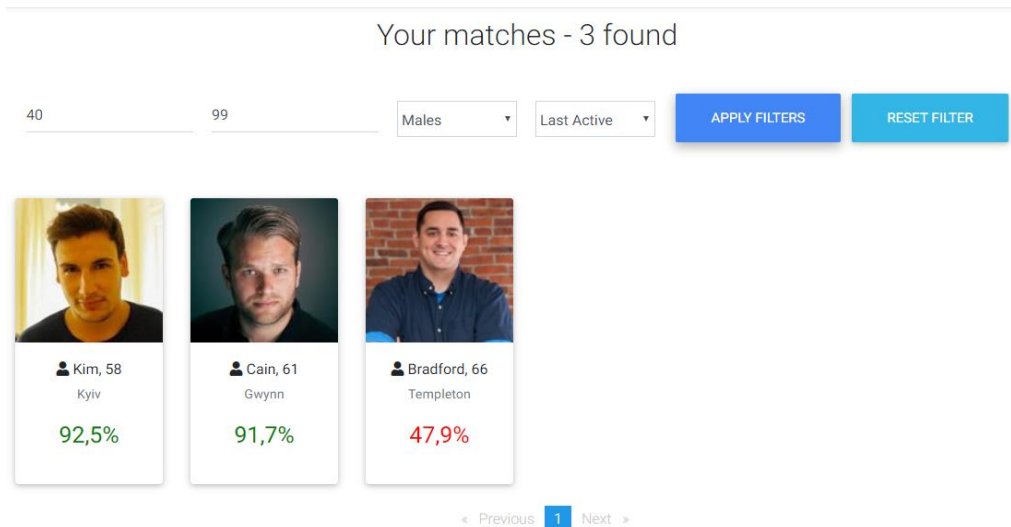


Figure 46: List of filters

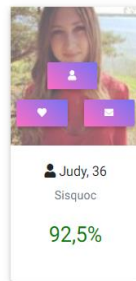


Figure 47: User selection

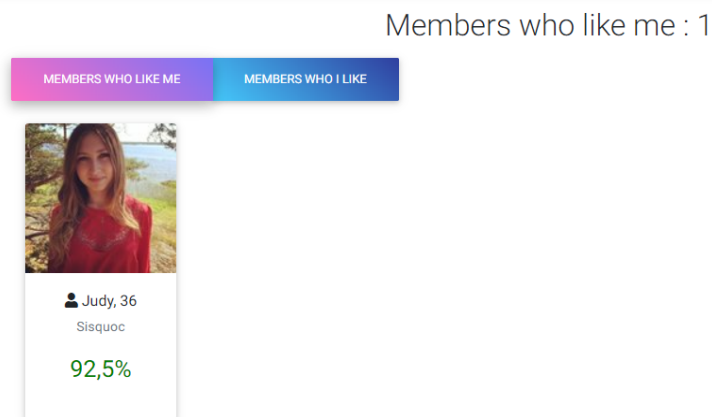


Figure 48: Users who chose us

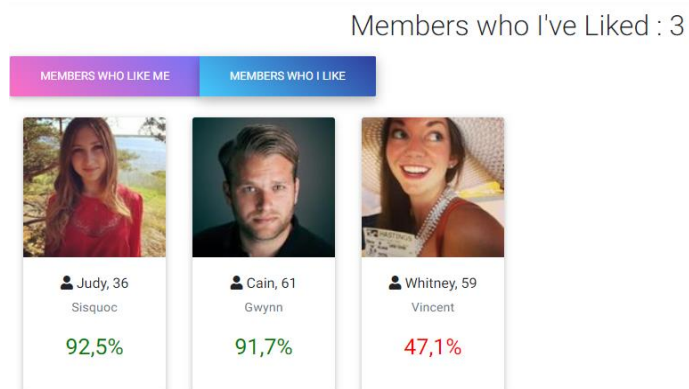


Figure 49: Users that we have selected

Fig. 50-52 shows the selected user's basic profile information, a tab with the user's interests, and a tab with all the user's photos.

### Judy's Page

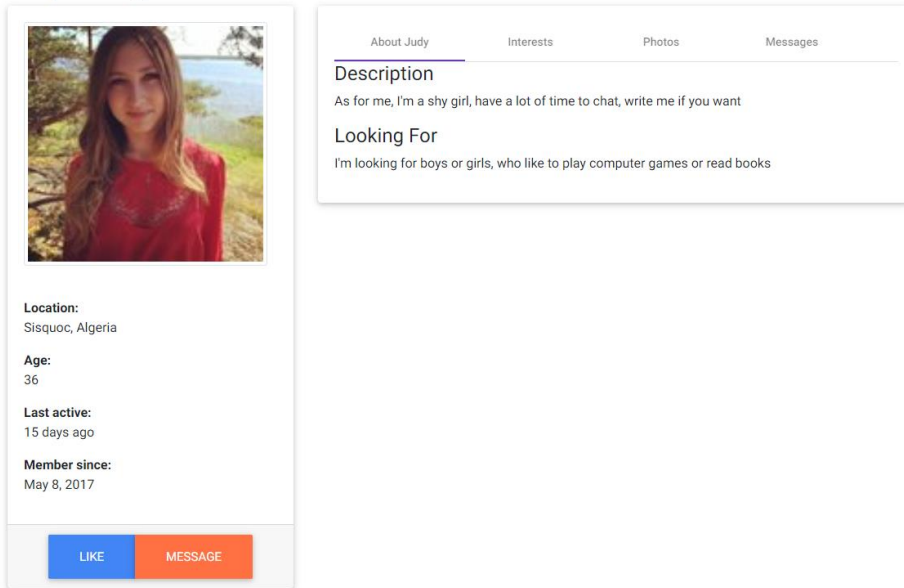


Figure 50: Main information about user

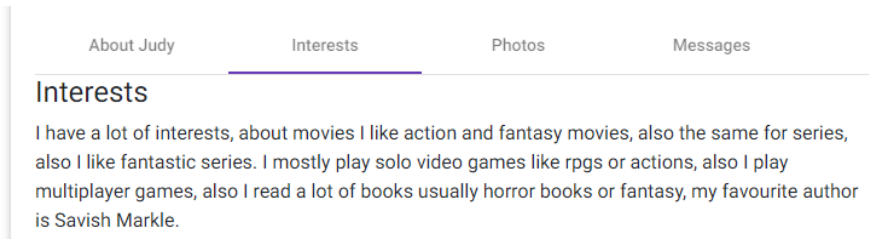


Figure 51: User interests

### Judy's Page

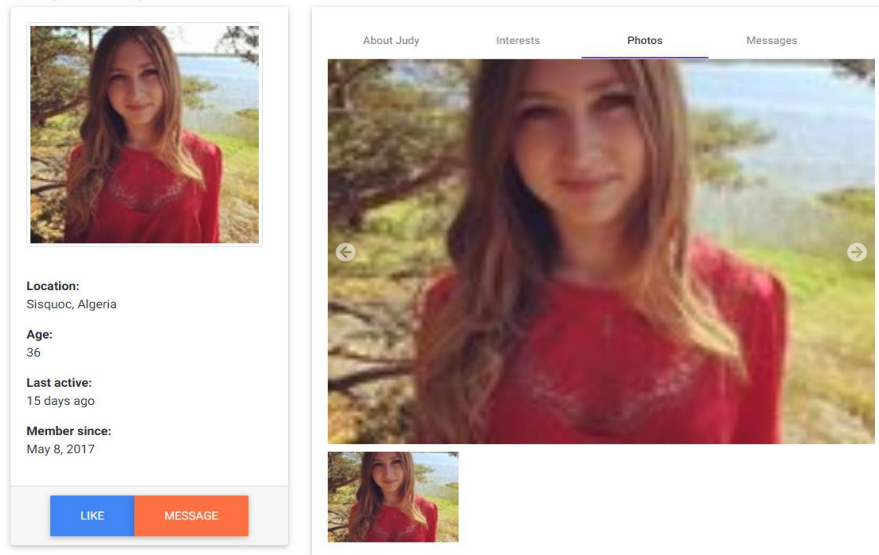


Figure 52: User photos

Fig. 53 shows a tab of private correspondence with the user. It offers user nicknames, photos, time of sending and reading messages. Fig. 54-55 demonstrate a page with information about all messages unread, received, and sent. You can manage notifications, namely, view the selected message by tracing a dialogue with the user, or delete the selected message for everyone or only for yourself.

# Judy's Page

The screenshot shows a user profile for 'Judy' on the left and a 'Messages' tab on the right. The profile includes a photo, location (Sisquoc, Algeria), age (36), last active status (15 days ago), and member since date (May 8, 2017). The messages tab shows a conversation with 'Bruce' where Judy says 'Hi, I'm ok' and Bruce replies 'how are you' and 'hi there'. A 'SEND' button is visible at the bottom of the messages area.

Figure 53: Private correspondence with the user

This screenshot shows a list of received messages. At the top are buttons for 'UNREAD', 'INBOX', and 'OUTBOX'. The message list has columns for 'Message', 'From / To', and 'Sent / Received'. Two messages are shown: 'Nice to meet you!' and 'Hi, I'm ok', both from 'Judy' and received '16 days ago'. Each message has a 'DELETE' button. A pagination bar at the bottom shows 'Previous 1 Next'.

Figure 54: Received messages

This screenshot shows a list of sent messages. At the top are buttons for 'UNREAD', 'INBOX', and 'OUTBOX'. The message list has columns for 'Message', 'From / To', and 'Sent / Received'. Three messages are shown: ':)', 'how are you', and 'hi there', all sent to 'Judy' '16 days ago'. Each message has a 'DELETE' button. A pagination bar at the bottom shows 'Previous 1 Next'.

Figure 55: Sent messages

Fig. 56-57 show logging in from another user's profile selected by the first user of the system and viewing the list of users who put a like mark. It allows you to start a private correspondence. Fig. 58 shows a personal correspondence between the registered user and the one selected by the system.

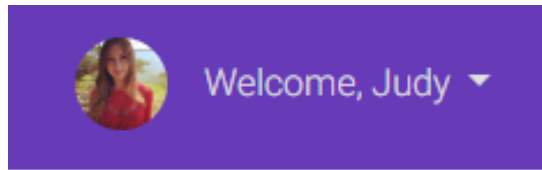


Figure 56: Another user's login to the system

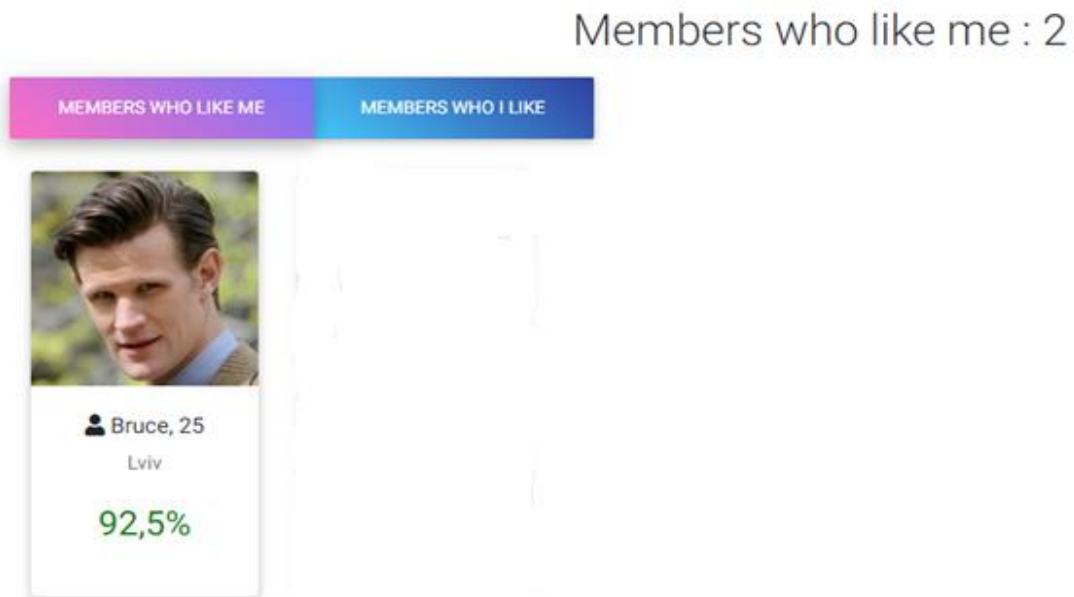


Figure 57: Users who selected the current user

### Bruce's Page

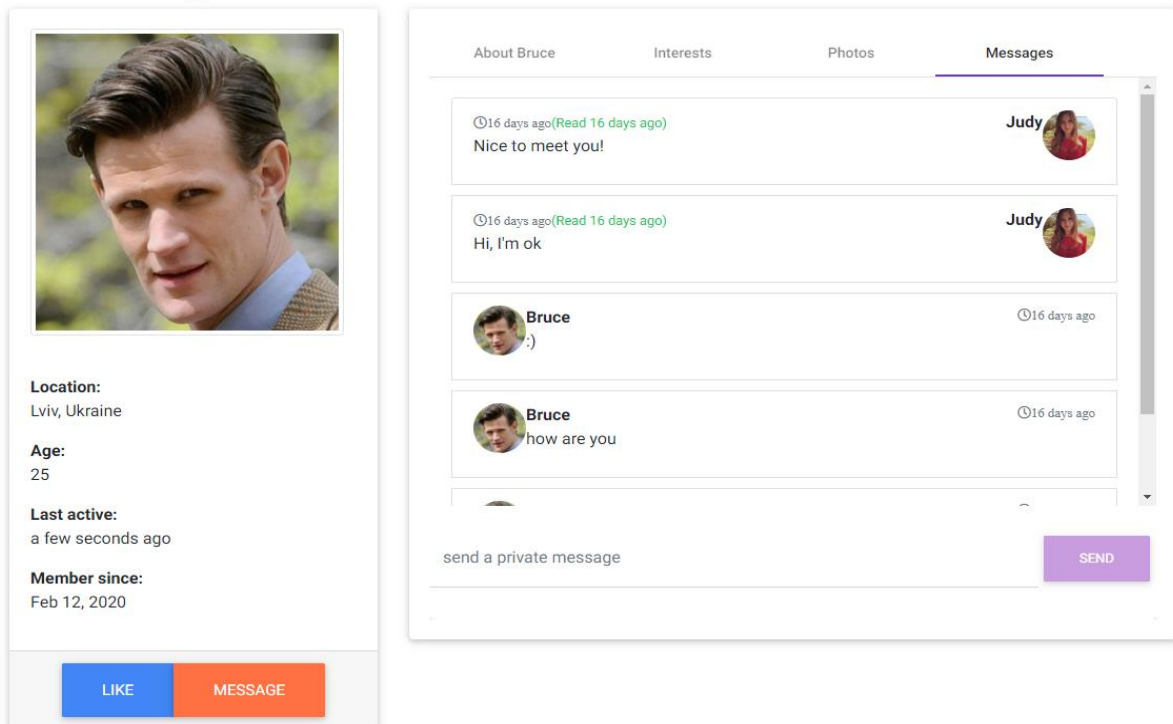


Figure 58: Private correspondence

After receiving the implemented intelligent system, a statistical analysis of two system parameters was carried out, namely, comparing the speed of user sample formation and the accuracy of the obtained percentage ratio. The combination of Levenshtein algorithms, N-gram, sample extension, Noisy Channel model, and the standard Levenshtein algorithm is most often used in similar systems for the socialization of individuals, being implemented they are respectively compared.

First, we analyzed the efficiency of forming a sample of users and selected twelve system users. Then we created a selection for each one using the Levenshtein algorithm and a combination of algorithms. The diagram shown in Fig. 59 demonstrates that the combination of algorithms is more efficient and accurate by about 25-30% comparing with the usual Levenshtein one. Next, the speed of forming a sample of users was analyzed. We again selected twelve users of the system and developed an example using a combination of algorithms and the standard Levenshtein algorithm. The resulting diagram is shown in Fig. 60. It can be concluded that the combination of algorithms implemented in the system makes the sample about ten times faster than the standard Levenshtein algorithm.

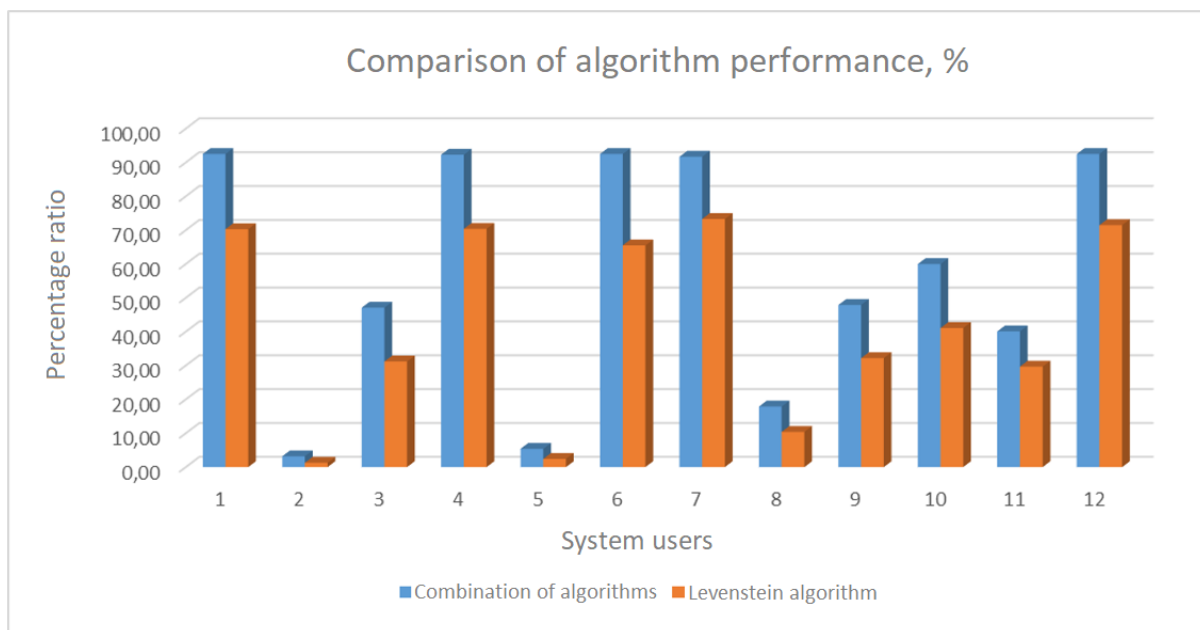


Figure 59: Percent ratio between users

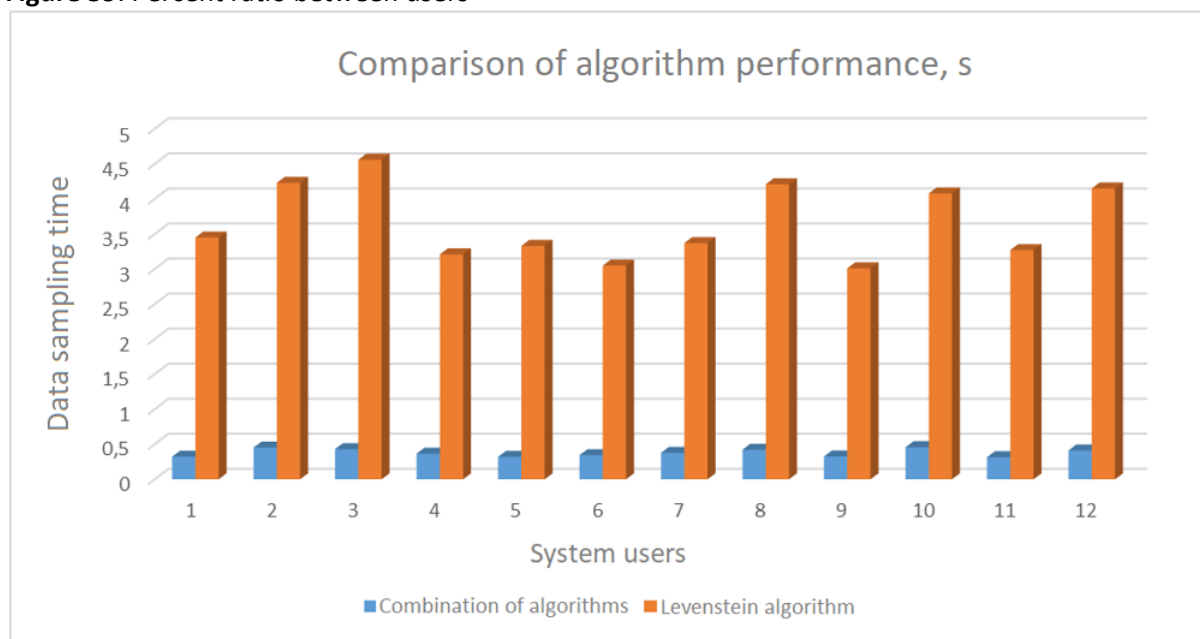


Figure 60: Time of formation data sampling

During the practical implementation of the user socialization system with shared interests, the created software tool, all algorithms, and system functions were described. All the features and procedures for performing the user socialization process were highlighted. A user manual was also created to provide basic information about the designed intelligent system. We also analyzed a control sample in which the main steps of performing the main algorithms and functions of the created intelligent system were demonstrated. The main goal of the work is to develop a smart strategy for the socialization of individuals with shared interests based on SEO technologies and machine learning methods. It is developed using fuzzy word search algorithms, the Noisy Channel model with algorithms for efficient distribution of text information, and a convolutional neural network to identify users of the system. No analogue system exists to analyze the specified information about the user and form a list of the most relevant users. Creating an intelligent system for the socialization of individuals is an urgent task since, in the modern world, people try to optimize all life processes to save time. When searching for specific programs, users first choose those that save time, optimize their work, and are automated enough to perform most actions instead of the system user. Our intelligent system combines two critical tasks: socialize users, maximize and automate the socialization process itself.

The design of a budget assessment will show that the development of this software product is profitable. The created intelligent system offers unique user search and analysis opportunities, and large software development companies will be interested in buying this software product. Marketing assessment reveals that the system meets the users' needs as it includes functionality that is not available in similar software products. It is easy and convenient to use and offers an entirely new approach to searching, analyzing, and socializing users within the system.

Currently, the software market is developing rapidly, and new software products are constantly emerging that affect users' lives using the created intelligent systems. The most popular are browser-based web applications, applications for smartphones implemented using a client-server data transfer model, ongoing working online programs frequently used by different users. It embraces perspectives for further development of various software products. Similar products on the market include all existing systems designed for finding and socializing users, which are usually free to use but have paid subscriptions. Larger companies purchase the intelligent systems themselves to expand their software products. The primary users are the people whose main goal is to find new friends with similar interests. Also, this search should be fast and easy to use, automated to search for people, and effectively save time. There are many similar programs, but there are several competing systems that have identical functionality. Among the programs found, it is worth highlighting three analogues similar to the intelligent system being created. They are Tinder, Badoo, and Chatous programs. If we consider each of these programs separately, we can identify the main advantages and disadvantages that need to be considered when creating a system. In analyzing all the available information, we can assume an excellent opportunity to create a new software product aimed at the mass user since everyone will search for people with shared interests. Also, the designed product offers options that analogues do not possess, which makes it competitive and unique among similar programs in the market.

## **7. Conclusion**

In our time, the socialization of individuals according to shared interests is an essential process since most people try to simplify and automate the main life processes, which usually take up a lot of free time. The same applies to socialization, and the developed intelligent system plays an essential role in this regard. In performing the work, the top scientific results related to this topic and play a significant role in forming an understanding of the main implementation processes of the intelligent system of socialization according to shared interests were analyzed. We investigated the main algorithms that play an essential role in social networks and various user interaction systems. Algorithms that are best suited for the current intelligent system were identified. The use of modern neural networks and their importance in automating the operation of an intelligent system in data processing issues were investigated. While studying the system analysis, a decision tree was built to outline the essence of the intelligent system and determine the primary purpose of the system's functioning. The intelligent system structure was also specified using IDEF0 and IDEF3 diagrams, and the hierarchy of system processes was constructed using a tree structure.

Choosing the right software solutions for writing an intelligent system is essential since most of its features depend. We worked with the programming languages C# and JavaScript. Their main frameworks ASP .NET Core and Angular, together with the corresponding software code development environments such as Visual Studio and Visual Studio Code, made it possible to write a multi-level system. It is divided into server and client parts, dynamically and completely asynchronously interact with each other during the system work. The practical implementation of the intelligent system of the socialization of individuals with shared interests is the most crucial step. Reliable and straightforward functionality of registration and authorization using identity and JWT tokens methods were developed. It helped securely store user passwords in the database, optimize the creation of a session, and provide all the necessary functionality while working in the system. Next, two neural networks were implemented: convolutional and Siamese, which allowed us to search for a human face in the photos that the user uploads and compare the found face with the available faces in the database. It helped us effectively identify the user's authenticity and guarantee that this user is currently not in the database, so he is real. An algorithm for analyzing and comparing user information was created using fuzzy search algorithms, the Levenshtein algorithm, and the Noisy Channel model. It generates a list of available users of the system, sorted in descending order of the percentage of user similarity. It indicates how much the interests of other users coincide with the interests of the current user. The algorithm of sampling of users of similar interests implemented in the IS is 25-30% more efficient and accurate than usual Levenshtein algorithm. At the same time, the implemented algorithm performs sampling 10 times faster than Levenshtein algorithm. We achieved our goals and objectives in work, but there are still many things that can be improved inside the created intelligent system. Still, the functionality available inside the intelligent system corresponds to the goal of its creation. All system users can register, including personal data, upload photos with their faces and then carry out adequate socialization. It includes viewing users from the generated list, checking the user's preferences, and conducting private correspondence within the system with the selected user. Finally, a discussion of the results follows, highlighting the most important findings and their interpretation, the data limitations, and the perspective for further research. Our findings put forward a theory that gives a strong perspective for further research and development of new inexpensive catalysts with superior CO tolerance and durability. While appreciating the immense contributions that such studies have made to the field, it is imperative to note that there is still the need to bring the various dimensions of research collaboration into a holistic perspective for further research and policy attention, particularly where research collaboration is a crucial aspect of a nation's economic development strategy.

## 8. References

- [1] S. C. Chu, Using a consumer socialization framework to understand electronic word-of-mouth (eWOM) group membership among brand followers on Twitter, *Electronic Commerce Research and Applications* 14(4) (2015) 251–260. doi: 10.1016/j.elerap.2015.04.002.
- [2] F. De-Gregorio, Y. Sung, Understanding attitudes toward and behaviors in response to product placement, *Journal of Advertising* 39(1) (2010) 83–96. doi: 10.2753/JOA0091-3367390106.
- [3] S. M. Elaheebocus, M. Weal, L. Morrison, Peer-based social media features in behavior change interventions: Systematic review, *Journal of Medical Internet Research* 20(2) (2018) 1–20. doi: 10.2196/jmir.8342.
- [4] I. Erkan, The influence of e-WOM in social media on consumers' purchase intentions: An extended approach to information adoption, *Computers in Human Behavior* 4 (2016) 47–55. doi: 10.1016/j.chb.2016.03.003.
- [5] E. Ferrara, R. Interdonato, A. Tagarelli, Online popularity and topical interests through the lens of Instagram, in: *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, 2, 2014, pp. 24–25. doi: 10.1145/2631775.2631808.
- [6] L. Gao, X. Bai, Online consumer behavior and its relationship to website atmospheric induced flow: Insights into online travel agencies in China, *Journal of Retailing and Consumer Services* 21(4) (2014) 653–655. doi: 10.1016/j.jretconser.2014.01.001.

- [7] A. N. Geurin-Eagleman, L. M. Burch Communicating via photographs: A gendered analysis of Olympic athletes' visual self -presentation on Instagram, *Sport Management Review* 19(2) (2015) 133–145. doi: 10.1016/j.smr.2015.03.002.
- [8] J. D. Guidry, M. Messner, Y. Jin, From McDonalds fail to Dominos sucks: An analysis of Instagram images about the 10 largest fast food companies, *Corporate Communications: An International Journal* 20(3) (2015) 344–359. doi: 10.1108/CCIJ-04-2014-0027.
- [9] R. Hanna, A. Rohm, V. L. Crittenden, We're all connected: The power of the social media ecosystem, *Business Horizons* 54(3) (2011) 265–273. doi: 10.1016/j.bushor.2011.01.007.
- [10] M. Salganik, *Bit by bit: Social research in the digital age*. Princeton University Press 2019.
- [11] S. Hudson, M. S. Roth, T. J. Madden, R. Hudson, The effects of social media on emotions, brand relationship quality, and word of mouth: An empirical study of music festival attendees, *Tourism Management* 2(8) (2015) 68–76. doi: 10.1016/j.tourman.2014.09.001.
- [12] J. McCarthy, J. Rowley, C. J. Ashworth, E. Pioch, Managing brand presence through social media: The case of UK football clubs, *Internet Research* 24(2) 2014 (181–204). doi: 10.1108/IntR-08-2012-0154.
- [13] E. Kim, Y. Sung, H. Kang, Brand followers' retweeting behaviour on Twitter: How brand relationship influence brand electronic word-of-mouth, *Computers in Human Behavior* 38(8) (2014) 18–25. doi: 10.1016/j.chb.2014.04.020.
- [14] C. Kudeshia, P. Sikdar, A. Mittal, Spreading love through fan page liking: A perspective on small scale entrepreneurs, *Computers in Human Behavior* 8(19) (2016) 257–270. doi: 10.1016/j.chb.2015.08.003.
- [15] J. E. Lueg, R. Z. Finney, Interpersonal communication in the consumer socialization process: Scale development and validation, *Journal of Marketing Theory and Practice* 15(1) (2007) 25–39. doi: 10.2753/MTP1069-6679150102.
- [16] M. Mousavijad, S. Payvandi, The effect of socialization factors on decision making of teenagers consumers in schools, *Journal of School Administration* 5(1) (2017) 217–234.
- [17] R. Schnell, Enhancing Surveys with Objective Measurements and Observer Ratings, *Journal of Interactive Marketing* 44(1) (2015) 288–302.
- [18] M. E. Parry, T. Kawakami, K. Kishiya, The effect of personal and virtual word-of-mouth on technology acceptance, *Journal of Product Innovation Management* 29(6) (2012) 952–966. doi: 10.1111/j.1540-5885.2012.00972.x.
- [19] A. Quan-Haase, L. Sloan, Introduction to the Handbook of Social Media Research Methods: Goals, Challenges and Innovations, *The Sage Handbook of Social Media Research Methods* 10(5) (2017) 606–859.
- [20] S. T. Murphy, R. B. Zajonc, Affect, cognition, and awareness: Affective priming with optimal and suboptimal stimulus exposures, *Journal of Personality and Social Psychology* 8(3) (2001) 723–739. doi: 10.1037/0022-3514.64.5.723.
- [21] M. Ngoma, P. D. Ntale, Word of mouth communication: A mediator of relationship marketing and customer loyalty, *Cogent Business and Management* 6(18) (2019) 3–6. doi: 10.1080/23311975.2019.1580123.
- [22] A. Ozdemir, B. Tozlu, E. Şen, A. Ateşoğlu, Analyses of word-of-mouth communication and its effect on students' university preferences, *Procedia – Social and Behavioral Sciences* 8(5) (2016) 22–35. doi: 10.1016/j.sbspro.2016.11.022.
- [23] J. Park, G. L. Ciampaglia, F. Ferrara, Style in the age of Instagram: Predicting success within the fashion industry using social media, in: *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, 22(8), 2016, pp. 64–72. doi: 10.1145/2818048.2820065.
- [24] V. Lytvyn, V. Danylyk, M. Bublyk, L. Chyrun, V. Panasyuk, O. Korolenko, The lexical innovations identification in english-language eurointegration discourse for the goods analysis by comments in e-commerce resources, in: *Proceedings of IEEE 16th International conference on Computer science and information technologies*, Lviv, Ukraine, 2021, pp. 85–97. doi: 10.1109/CSIT52700.2021.9648594.
- [25] B. Ranjbaran, M. Jamshidian, Z. Dehghan, A survey of identification of major factors influencing customers attitude toward machine made carpet brands, *Journal of Business Strategies* 5(23) (2007) 109–118.



- [26] T. Batiuk, V. Vysotska, V. Lytvyn, Intelligent system for socialization by personal interests on the basis of SEO technologies and methods of machine learning, CEUR workshop proceedings Vol-2604 (2020) 1237–1250.
- [27] M. Komar, V. Golovko, A. Sachenko, S. Bezobrazov, Intelligent system for detection of networking intrusion, in: Proceedings of the 6th IEEE Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems, 2011, pp. 374–377, doi: 10.1109/IDAACS.2011.6072777.
- [28] B. Rusyn, L. Pohreliuk, A. Rzhenskyi, R. Kubik, Y. Ryshkovets, L. Chyrun, S. Chyrun, A. Vysotskyi, V.B. Fernandes, The mobile application development based on online music library for socializing in the world of bard songs and scouts' bonfires, Advances in Intelligent Systems and Computing 1080 (2020) 734–756. doi: 10.1007/978-3-030-33695-0\_49.
- [29] A. Batiuk, V. Voityshyn, Apache storm based on topology for real-time processing of streaming data from social networks, in: Proceedings of the 2016 IEEE 1st International Conference on Data Stream Mining and Processing, DSMP, 2016, pp. 345–349. doi: 10.1109/DSMP.2016.7583573.
- [30] H. Lipyana, V. Maksymovych, A. Sachenko, T. Lendyuk, A. Fomenko, I. Kit, Assessing the Investment Risk of Virtual IT Company Based on Machine Learning, Data Stream Mining and Processing, 2020, pp. 167–187. doi: 10.1007/978-3-030-61656-4\_11.
- [31] A. Peleshchyshyn, O. Markovets, V. Vus, S. Albota, Identifying specific roles of users of social networks and their influence methods, in: Proceedings of the IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 2, 2018, pp. 39–42. doi: 10.1109/STC-CSIT.2018.8526635.
- [32] A. Peleshchyshyn, O. Mastyakash, Analysis of the methods of data collection on social networks, in: Proceedings of the Conference on Computer Sciences and Information Technologies, CSIT, 1, 2017, pp. 175–178. doi: 10.1109/STC-CSIT.2017.8098762.
- [33] O. Anisimova, V. Vasylenko, S. Fedushko, Social networks as a tool for a higher education institution image creation, CEUR Workshop Proceedings Vol-2392 (2019).
- [34] A. Peleshchyshyn, V. Vus, S. Albota, O. Markovets, A Formal Approach to Modeling the Characteristics of Users of Social Networks Regarding Information Security Issues, Advances in Intelligent Systems and Computing 902 (2020) 485–494. doi: 10.1007/978-3-030-12082-5\_44.
- [35] R. Hryshchuk, K. Molodetska, Y. Syerov, Method of improving the information security of virtual communities in social networking services, CEUR Workshop Proceedings 2392 (2019) 23–41.
- [36] A. Bomba, N. Kunanets, V. Pasichnyk, Y. Turbal, Model of the information shock waves in social network based on the special continuum neural network, in: Proceedings of IEEE 1st International Conference on System Analysis and Intelligent Computing, SAIC, 2018. doi: 10.1109/SAIC.2018.8516809.
- [37] A. Bomba, N. Kunanets, V. Pasichnyk, Y. Turbal, Mathematical and computer models of message distribution in social networks based on the space modification of Fermi-Pasta-Ulam approach, Advances in Intelligent Systems and Computing 836 (2019) 257–266. doi: 10.1007/978-3-319-97885-7\_26.
- [38] A. Bomba, N. Kunanets, V. Pasichnyk, Y. Turbal, Process modeling of message distribution in social networks based on socio-communicative solitons, International Journal of Computing 17(4) (2018) 250–259.
- [39] K. Sloboda, The peculiarities of communicative process in Social Networks, in: Proceedings of the 10th International Conference on Experience of Designing and Application of CAD Systems in Microelectronics, CADSM, 2009, pp. 553–555.
- [40] A. Peleshchyshyn, H. Bandrovskyi, Informational influence in social networks: Dynamics modeling based on the system of linear equations, in: Proceedings of IEEE 14th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 1, 2019, pp. 165–168. doi: 10.1109/STC-CSIT.2019.8929819.
- [41] A. Peleshchyshyn, T. Klynina, S. Gnatyuk, Legal mechanism of counteracting information aggression in social networks: From theory to practice, CEUR Workshop Proceedings Vol-2392 (2019).
- [42] A. Bomba, N. Kunanets, V. Pasichnyk, Yu. Turbal, M. Nazaruk, Software-algorithmic tool for analyzing the processes of messages distribution in social networks, CEUR Workshop Proceedings Vol-2387 (2019) 426–436.

- [43] Y. Butelskyy, Statistical Methods to Detect Gender Peculiarities of Communication in Vkontakte Social Network Groups, in: Proceedings of the 11th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 2016, pp. 132–135. doi: 10.1109/STC-CSIT.2016.7589888.
- [44] Y. Ulyana, P. Andrij, Conceptual model the speed of information dissemination in social networks on the basis of the Monod equation, in: Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 1, 2017, pp. 318–321. doi: 10.1109/STC-CSIT.2017.8098795.
- [45] N. Vovk, M. Klymash, Problem-oriented social network for the exchange of international professional experience, CEUR Workshop Proceedings Vol-2392 (2019) 252–262.
- [46] T. Yavorska, O. Prihunov, Y. Syerov, Libraries in social networks: Opportunities and presentations, CEUR Workshop Proceedings Vol-2392 (2019) 242-251.
- [47] I. Davydova, O. Marina, A. Slianyk, Y. Syerov, Social networks in developing the internet strategy for libraries in Ukraine, CEUR Workshop Proceedings Vol-2392 (2019) 122–133.
- [48] D. Tarasov, Z. Koval, M. Klymash, Efficiency evaluation of using social networks application in the university e-learning system, CEUR Workshop Proceedings Vol-2392 (2019) 12-22.
- [49] A. Petrushka, M. Komova, I. Demydov, Social networks as an alternative environment for the implementation of scientific communication, CEUR Workshop Proceedings 2392 (2019) 79-94.
- [50] K. Mykich, Y. Burov, Research of uncertainties in situational awareness systems and methods of their processing, Eastern-European Journal of Enterprise Technologies 1(4) (2016) 19–27. doi: 10.15587/1729-4061.2016.60828.
- [51] B. Polishchuk, A. Berko, L. Chyrun, M. Bublyk, V. Schuchmann, The rain prediction in Australia based Big Data analysis and machine learning technology, in: Proceedings of IEEE 16th International conference on computer science and information technologies, Lviv, Ukraine, 2021, pp. 97–100. doi: 10.1109/CSIT52700.2021.9648691.
- [52] K. Mykich, Y. Burov, Algebraic model for knowledge representation in situational awareness systems, in: Proceedings of the 11th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 2016, pp. 165–167. doi: 10.1109/STC-CSIT.2016.7589896.
- [53] S. Makara, L. Chyrun, Y. Burov, Z. Rybchak, I. Peleshchak, R. Peleshchak, R. Holoshchuk, S. Kubinska, A. Dmytriv, An intelligent system for generating end-user symptom recommendations based on machine learning technology, CEUR Workshop Proceedings 2604 (2020) 844–883.
- [54] V. Lytvyn, I. Peleshchak, R. Peleshchak, The compression of the input images in neural network that using method diagonalization the matrices of synaptic weight connections, in: Proceedings of the 2nd International Conference on Advanced Information and Communication Technologies, AICT, 2017, pp. 66–70. doi: 10.1109/AIACT.2017.8020067.
- [55] V. Khavalko, I. Tsmots, Image classification and recognition on the base of autoassociative neural network usage, in: Proceedings of IEEE 2nd Ukraine Conference on Electrical and Computer Engineering, UKRCON, 2019, pp. 1118–1121. doi: 10.1109/UKRCON.2019.8879774.
- [56] R. Tushnytskyy, I. Perevalova, G. Pavlyuk, Web content-based image retrieval system based on 4-D structural features, in: Proceedings of the 10th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science, TCSET, 2010, p. 281.
- [57] R. Melnyk, R. Tushnytskyy, Pattern keys investigation for content-based image retrieval system, in: Proceedings of the 10th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science, TCSET, 2010, p. 94.
- [58] R. Melnyk, R. Tushnytskyy, 4-D pattern structure features by three stages clustering algorithm for image analysis and classification, Pattern Analysis and Applications 16(2) (2013) 201–211. doi: 10.1007/s10044-013-0326-x
- [59] R. Melnyk, R. Tushnytskyy, Circuit board image analysis by clustering, in: Proceeding of the 4th International Conference of Young Scientists on Perspective Technologies and Methods in MEMS Design, MEMSTECH, 2008, pp. 44–45. doi:10.1109/MEMSTECH.2008.4558732.
- [60] O. Veres, B. Rusyn, A. Sachenko, I. Rishnyak, Choosing the method of finding similar images in the reverse search system, CEUR Workshop Proceedings 2136 (2018) 99–107.

- [61] D. Peleshko, T. Rak, I. Izonin, Image superresolution via divergence matrix and automatic detection of crossover, *International Journal of Intelligent Systems and Applications* 8(12) (2016) 1–8. doi: 10.5815/ijisa.2016.12.01.
- [62] R. Tkachenko, P. Tkachenko, I. Izonin, Y. Tsymbal, Learning-based image scaling using neural-like structure of geometric transformation paradigm, *Studies in Computational Intelligence* 730 (2018) 537–565. doi: 10.1007/978-3-319-63754-9\_25.
- [63] Y. Rashkevych, D. Peleshko, O. Vynokurova, I. Izonin, N. Lotoshynska, Single-frame image super-resolution based on singular square matrix operator, in: *Proceedings of IEEE 1st Ukraine Conference on Electrical and Computer Engineering, UKRCON, 2017*, pp. 944–948. doi: 10.1109/UKRCON.2017.8100390.
- [64] N. Shakhovska, L. Nych, R. Kaminskyj, The identification of the operator's systems images using the method of the phase portrait, *Advances in Intelligent Systems and Computing* 512 (2017) 241–253. doi: 10.1007/978-3-319-45991-2\_16.
- [65] B. Rusyn, I. Prudyus, V. Ostap, Fingerprint image enhancement algorithm, in: *Proceedings of the 6th International Conference on The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM, 2001*, pp. 193–194. doi: 10.1109/CADSM.2001.975804.
- [66] M. Nazarkevych, R. Oliarnyk, S. Dmytruk, An images filtration using the Ateb-Gabor method, in: *Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, 2017*, pp. 208–211. doi: 10.1109/STC-CSIT.2017.8098770.
- [67] M. Nazarkevych, I. Klyujnyk, I. Maslanych, B. Havrysh, H. Nazarkevych, Image filtration using the Ateb-Gabor filter in the biometric security systems, in: *Proceedings of 14th International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH, 2018*, pp. 276–279. doi: 10.1109/MEMSTECH.2018.8365749.
- [68] A. Kovalchuk, N. Lotoshynska, Elements of RSA Algorithm and Extra Noising in a Binary Linear-Quadratic Transformations during Encryption and Decryption of Images, in: *Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP, 2018*, pp. 542–544. doi: 10.1109/DSMP.2018.8478471.
- [69] O. Riznik, I. Yurchak, E. Vdovenko, A. Korchagina, Model of stegosystem images on the basis of pseudonoise codes, in: *Proceedings of the 6th International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH, 2010*, pp. 51–52.
- [70] M. Nazarkevych, O. Riznyk, V. Samotyy, U. Dzelendzyak, Detection of regularities in the parameters of the Ateb-Gabor method for biometric image filtration, *Eastern-European Journal of Enterprise Technologies* 1(2-97) (2019) 57–65. doi: 10.15587/1729-4061.2019.154862.
- [71] O. Riznyk, Y. Kynash, O. Povshuk, Y. Noga, The method of encoding information in the images using numerical line bundles, in: *Proceedings of IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 1, 2018*, pp. 80–83. doi: 10.1109/STC-CSIT.2018.8526751.
- [72] I. Ruban, H. Khudov, V. Khudov, I. Khizhnyak, O. Makoveichuk, Segmentation of the images obtained from onboard optoelectronic surveillance systems by the evolutionary method, *Eastern-European Journal of Enterprise Technologies* 5(9-89) (2017) 49–57. doi: 10.15587/1729-4061.2017.109904.
- [73] E. Yelmanova, Quantitative evaluation of contrast for a complex image by its histogram, in: *Proceedings of the 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science, TCSET, 2016*, pp. 688–692. doi: 10.1109/TCSET.2016.7452155.
- [74] M. Nazarkevych, B. Yavourivskiy, I. Klyuynyk, Editing raster images and digital rating with software, in: *Proceedings of 13th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM, 2015*, pp. 439–441. doi: 10.1109/CADSM.2015.7230897.
- [75] I. Prudyus, S. Voloshynovskiy, A. Synyavskyy, Wavelet-based MAP image denoising using provably better class of stochastic i.i.d. image models, in: *Proceedings of 5th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Service, TELSIS, 2, 2001*, pp. 583–586. doi: 10.1109/TELSKS.2001.955843.

- [76] E. S. Yelmanova, Y. M. Romanyshyn, No-reference contrast metric for medical images, in: Proceedings of IEEE 37th International Conference on Electronics and Nanotechnology, ELNANO, 2017, pp. 338–343. doi: 10.1109/ELNANO.2017.7939774.
- [77] O. Kosar, N. Shakhovska, An Overview of Denoising Methods for Different Types of Noises Present on Graphic Images, *Advances in Intelligent Systems and Computing* 871 (2019) 38–47. doi: 10.1007/978-3-030-01069-0\_4.
- [78] I. Dronyuk, O. Fedevych, B. Demyda, Signals and Images Protection Based on Ateb-Transforms in Infocommunication Systems, in: Proceedings of International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PICSandT, 2019, pp. 461–464. doi: 10.1109/INFOCOMMST.2018.8632043.
- [79] B. Rusyn, R. Kosarevych, O. Lutsyk, V. Korniy, Segmentation of atmospheric cloud images by remote sensing, in: Proceedings 14th international conference TCSET, 2018, pp.213–216. doi: 10.1109/TCSET.2018.8336189.
- [80] B. Rusyn, O. Lutsyk, R. Kosarevych, Evaluation the informativity of a training sample for image classification by deep learning methods, *Cybernetics and System Analysis* 57(6) (2021) 853–863. doi:10.1007/s10559-021-00411-4.
- [81] R. Kosarevych, O. Lutsyk, B. Rusyn, Detection of pixels corrupted by impulse noise using random patterns, in: *The Visual Computer* 2021. doi:10.1007/s00371-021-02207-1.
- [82] Z. Saaya, T. W. Hong. The development of trust matrix for recognizing reliable content in social media, *International Journal of Computing* 18(1) (2019) 60–66.
- [83] L. Chyrun, A. Gozhyj, I. Yevseyeva, D. Dosyn, V. Tyhonov, M. Zakharchuk, Web content monitoring system development, *CEUR Workshop Proceedings* 2362 (2019) 126-142.
- [84] V. Anand, C. Hota. Motivation of participants in crowdsourcing platforms using intelligent agents, *International Journal of Computing* 19(1) (2020) 78–87.
- [85] A. Dyryv, V. Andrunyk, Y. Burov, I. Karpov, L. Chyrun, The user's psychological state identification based on Big Data analysis for person's electronic diary, in: Proceedings of IEEE 16th International conference on computer science and information technologies, Lviv, Ukraine, 22–25 September, 2021, pp. 101–112. doi: 10.1109/CSIT52700.2021.9648810.
- [86] O. Semotiuk, Computer based content analysis: Main goals, applications, advantages and disadvantages, in: *The Experience of Designing and Application of CAD Systems in Microelectronics - Proceedings of the 7th International Conference, CADSM, 2003*, pp. 521–522. doi: 10.1109/CADSM.2003.1255139.
- [87] P. Zhezhnych, O. Markiv, Linguistic comparison quality evaluation of web-site content with tourism documentation objects, *Advances in Intelligent Systems and Computing* 689 (2018) 656–667. doi:10.1007/978-3-319-70581-1\_45.
- [88] O. Hladun, A. Berko, M. Bublyk, L. Chyrun, V. Schuchmann, Intelligent system for film script formation based on artbook text and Big Data analysis, in: Proceedings of IEEE 16th International conference on computer science and information technologies, Lviv, Ukraine, 2021, pp.138–146. doi: 10.1109/CSIT52700.2021.9648682.
- [89] I. Khomytska, V. Teslyuk, Modelling of phonostatistical structures of the colloquial and newspaper styles in English sonorant phoneme group, in: Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 2017, 1, pp. 67–70. doi: 10.1109/STC-CSIT.2017.8098738.
- [90] I. Khomytska, V. Teslyuk, Modelling of phonostatistical structures of English backlingual phoneme group in style system, in: Proceedings of 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM, 2017, pp. 324–327. doi: 10.1109/CADSM.2017.7916144.
- [91] I. Khomytska, V. Teslyuk, Authorship Attribution by Differentiation of Phonostatistical Structures of Styles, in: Proceedings IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 2018, 2, pp. 5–8. doi: 10.1109/STC-CSIT.2018.8526739.