

# Peculiarities of organizing the workspace using machine learning methods

Taras Basyuk <sup>a</sup>, Andrii Vasyliuk <sup>a</sup>, Lyubomyr Demkiv <sup>a</sup>

<sup>a</sup> Lviv Polytechnic National University, Bandera str.12, Lviv, 79013,

## Abstract

The article analyzes the space organization topic, defines its characteristics, and gives the factors that contribute to the conduct of this research. As a result of the research, an analysis of the subject area was carried out, and it was shown that today there is a limited set of tools (libraries) used in the process of pattern recognition, among the most famous are: DeepFace, TensorFlow and OpenCV. A description of the concepts of machine learning and recommender systems is given, which makes it possible to determine the main principles of their construction. A systematic analysis of the subject area was carried out and a tree of goals was built, which showed the subordination and interrelationship of goals in achieving the general goal. A mathematical description of the subject area was carried out using the algebra of algorithms, which provided the means to minimize created models by the number of uniterms. The presented models give a complete idea of the system construction features. The design of the software system was carried out using an object-oriented approach and the display of created diagrams in accordance with the notation of the UML language. The study presents diagrams of use cases and classes, which simplified the understanding of the information system functioning features for organizing the workspace using machine learning methods. The result of the research was the design and implementation of the system using a relational database and the Android Studio environment. The created software product works in prototype mode and implements the described functionality.

## Keywords <sup>1</sup>

Workspace, machine learning, algebra of algorithms, UML design, information system

## 1. Introduction

The workspace is where people spend a large part of their day. You can often hear complaints about how difficult it is to cope with tasks, and how much time and nerves it takes to look for something on the table or in drawers. You can hear the standard answers to this: it is fatigue, the need to rest, it is necessary to change the environment, work, etc. At the same time, the reason may be on the surface and consist in the fact that irritation may be caused by banal clutter or inconvenient organization of the workspace [1,2].

However, not everyone manages to easily organize their space, because not everyone has a developed spatial perception and creative thinking, which plays an important role in this process. A lot of information on organizing things can be found on the Internet [3], but the mentioned recommendations refer to general trends and well-known trends, which are not always relevant for a wide range of users. In view of that, the following main problems are highlighted in the process of organizing the workspace: considerable time consumption, understanding that there is an insufficient level of creativity and spatial vision, unwillingness to "understand" all the intricacies of organizing the space.

<sup>1</sup>MoMLeT+DS 2022: 4<sup>th</sup> International Workshop on Modern Machine Learning Technologies and Data Science, November, 25-26, 2022, Leiden-Lviv, The Netherlands-Ukraine

EMAIL: Taras.M.Basyuk@lpnu.ua (T. Basyuk); Andrii.S.Vasyliuk@lpnu.ua (A. Vasyliuk); lybomyr.i.demkiv@lpnu.ua (L. Demkiv)  
ORCID: 0000-0003-0813-0785 (T. Basyuk); 0000-0002-3666-7232 (A. Vasyliuk); 0000-0002-2802-3461 (L. Demkiv)



© 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)

## 1.1 Analysis of recent researches and publications

### 1.1.1. Domain analysis and known libraries

The conducted analysis showed that with all the variety of systems, there are currently no software applications that are focused on the workspace organization. And this is not surprising, since this task belongs to the process of pattern recognition, namely their identification and further presentation taking into account ergonomic features. To date, the optimal recognition algorithm has not been developed; there are several leading libraries that manage to recognize images on the pictures, search for identical individuals on the network. However, the goal of finding the location of the object or its identification was not achieved. The world's leading companies in the information technology field, such as Google, Microsoft, Facebook, Apple, Intel, have created departments for the image recognition libraries development [4,5]. In particular, FaceBook has developed an algorithm called DeepFace, which will allow visual analysis, comparison and identification of human faces with an accuracy of up to 97.25%. DeepFace uses a 3D modeling technique to scan an object, but the algorithm itself is built taking into account the "frontalization" process. Further, the obtained data are converted into numerical values and processed for further comparison [6].

Google has developed an open software library for machine learning called TensorFlow [7]. It allows solving the task of building and training a neural network for the purpose of automatically finding and classifying images, achieving the quality of human perception. While the reference implementation runs on single devices, TensorFlow can run on many parallel processors, both CPUs and GPUs, relying on the CUDA architecture to support general-purpose computing on GPUs) [8]. TensorFlow's computations are expressed as stateful dataflow graphs.

OpenCV is an open source library of computer vision, image processing, and general purpose numerical algorithms. Implemented in C/C++, also developed for Python, Java, Ruby, Matlab, Lua and other languages. Free for academic and commercial use - distributed under the terms of the BSD license. OpenCV provides various classifiers that can be used to recognize faces, eyes, cars, and many other objects. These classifiers, however, are quite simple, they are not trained using machine learning technologies, so when recognizing people, the accuracy will be approximately 80% [9].

Object recognition technologies have achieved high results, but they are insufficient for the everyday tasks implementation with the use of insignificant hardware resources. In this regard, it is necessary to improve object recognition algorithms and machine learning methods.

### 1.1.2. Application of machine learning methods in the subject area

Machine learning is a branch of artificial intelligence and computer science that focuses on using data and algorithms to mimic the way humans learn, gradually improving their accuracy. Machine learning refers to a class of computer algorithms that learn from examples rather than being explicitly programmed to perform a task. It learns to formulate a general rule from a set of concrete examples. Thus, just like human learning, the computer becomes able to improve its performance based on the acquired knowledge. The difference is that at the current state of our knowledge, a computer needs a lot more training examples than humans. There are three main parts in the machine learning algorithm [10]:

- decision making process: In general, machine learning algorithms are used for prediction or classification. Based on some input data, which may or may not be labeled, the algorithm will estimate the data pattern.
- error function: the error function is used to evaluate the prediction of the model. If there are known examples, the error function can make a comparison to assess the accuracy of the model.
- model optimization process: If the model can provide a better fit in the training set, then the weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this evaluation and optimization process, autonomously updating the weights until an accuracy threshold is reached.

The growing interest in machine learning is due to the same factors that made data mining and Bayesian analysis popular [11]. Namely, an increase in the amount and variety of available data, computational processing that is cheaper and more powerful, and affordable data storage. All of this means you can quickly and automatically build models that can analyze larger, more complex data and deliver faster, more accurate results. At the same time, by building accurate models, the organization has a better chance of identifying profitable opportunities or avoiding unknown risks [12].

As for this subject area, the application of machine learning methods will be carried out to create a recommender system. There are two main paradigms for building recommender systems - collaborative filtering and content filtering. Collaborative filtering is suitable for systems that are based on models of previous user behavior. The model can be built solely from the behavior of one user or (more efficiently) from the behavior of other users who share similar traits. Given the behavior of other users, collaborative filtering uses knowledge about groups to form recommendations based on shared traits. Essentially, the recommendations are based on the automatic collaboration of several users and filtering on those who show similar traits or behavior [13].

Another method of recommender systems is content-based filtering. In this method, a user profile is created to provide information about the types of items of interest to the user based on the keywords used to describe the items [14]. The content-based filtering method creates a profile for each element (based on a set of discrete attributes and features) that is used to characterize the element in the system. The system then creates a profile for the user based on a weighted vector of item characteristics (from items the user has previously rated or purchased and from items the user is viewing) [15]. The main problem with content-based filtering is that this method depends on the quality of elements and users' metadata description. This brings many limitations to the system that collaborative filtering helps to overcome. The last method does not depend on the quality and availability of metadata elements, but only on the behavior of users. Thus, it can be concluded that for the complete description of metadata elements systems, it is more appropriate to use content-based filtering, and for systems with a limited amount of data or its absence - collaborative filtering.

## **1.2 The main tasks of the research and their significance**

The purpose of the research is the design and implementation of organizing the workspace system using machine learning methods. The conducted research will provide the means to create a methodological basis in this direction, which will allow determining the strategy for the given task implementation.

To achieve the goal, the following main tasks must be solved: analyze the existing approaches, methods and software tools used in the field of organizing the workspace; to determine the main tasks that arise at the same time; carry out a systematic analysis of the subject area; conduct a mathematical description of the subject area using the algebra of algorithms; design the software system using an object-oriented approach and display the created diagrams in accordance with the UML language notation; to develop a software system for organizing the workspace using machine learning methods.

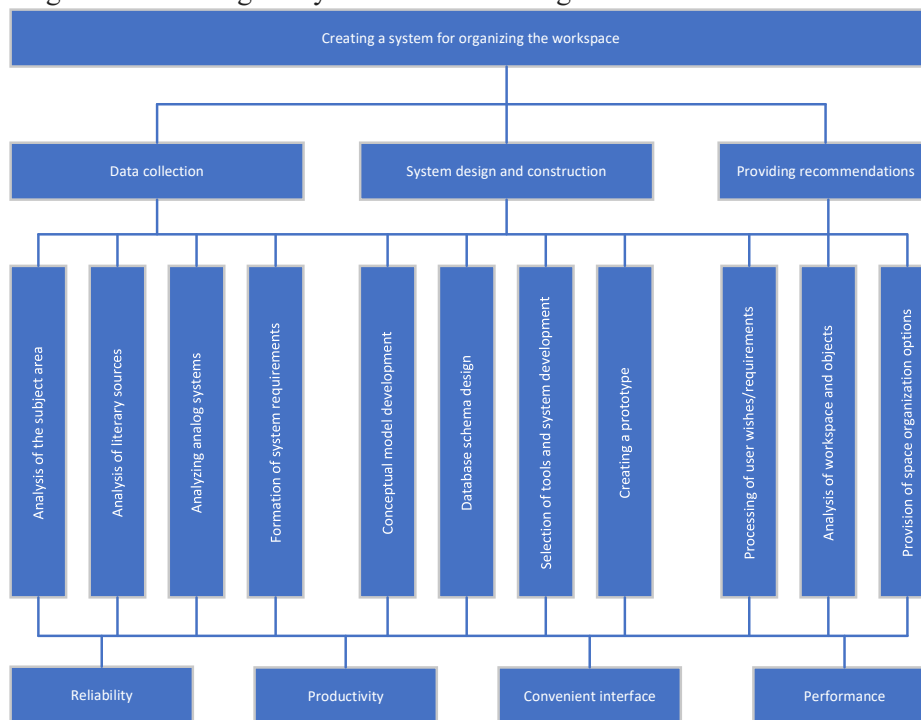
The results of the research solve the actual scientific and practical task of creating a methodological approach and software for solving the task of organizing the workspace.

## **2. Major research results**

At the initial stage of creating a system, it is necessary to carry out a system analysis, which is a sequence of actions to establish structural connections between the elements of the studied system [16]. For this, it is necessary to build a tree of goals. It is created to analyze a problematic situation and visualize its results. It consists of a general goal (the top of the tree, which shows what needs to be achieved) and sub-goals, the fulfillment of which will allow the main goal to be achieved. The general goal is: "Creating a system for organizing the workspace." Sub-goals have been defined to achieve the general goal:

- Data collection - a goal related to data analysis regarding system functionality and user interaction research. Includes:
  - carrying out analysis of the subject area;
  - analysis of literary sources;
  - analysis of systems of analogues;
  - formation of requirements for the designed system.
- System design and construction is a goal related to requirements visualization and system development. Includes:
  - development of a conceptual model;
  - database schema design;
  - selection of tools and system development;
  - creating a prototype.
- Providing recommendations is a goal related to satisfying user needs.
  - processing of user wishes/requirements;
  - analysis of workspace and objects;
  - providing of space organization options.
- System quality criteria:
  - reliability - the system's ability to work without failure during a specified period of time under specified conditions;
  - productivity - a criterion responsible for the speed of task performance;
  - convenient interface – a criterion responsible for the convenience of interaction with the user;
  - performance is a criterion responsible for the ability of the system to perform the assigned functions.

The tree of goals of the designed system is shown in Fig. 1.



**Figure 1:** Tree of goals

Further research was aimed at building models of system functioning using the algebra of algorithms apparatus. The first stage of the implementation of the algebra of algorithms was the description of uniterms and the synthesis of sequences [17], which is given below. Formed uniterms:  $I$  – uniterm for logging into the system;  $L$  is the unit term of authorization of an existing user;  $N$  – uniterm for registration of a new user;  $T$  is the uniterm of testing;  $S(s)$  is the space scanning uniterm;  $Rs(s)$  is the uniterm of repeated space scanning;  $S(o)$  – uniterm of object scanning;  $Rs(o)$  – uniterm of repeated

scanning of objects; *Rec* – a uniterm for forming a list of recommendations for organizing the workspace;  $u_1$  – condition for checking the absence of authentication data;  $u_2$  is the uniterm of the check for a successful scan of space,  $u_2$  is the uniterm of the check for a successful scan of objects. As a result of the use of the apparatus of the algebra of algorithms, the following sequences and eliminations were synthesized:

$S_1$  - sequence of system operation in case of creating a new user, successful scanning of space and objects:

$$S_1 = \overbrace{I, N, T, S(s), S(o), Rec}$$

$S_2$  - sequence of system operation in the case of creating a new user, successful scanning of space and unsuccessful scanning of objects:

$$S_2 = \overbrace{I, N, T, S(s), Rs(o), Rec}$$

$S_3$  – sequence of system operation in the case of creating a new user, unsuccessful scanning of space and successful scanning of objects:

$$S_3 = \overbrace{I, N, T, Rs(s), S(o), Rec}$$

$S_4$  – sequence of system operation in the case of creating a new user, unsuccessful scanning of space and unsuccessful scanning of objects:

$$S_4 = \overbrace{I, N, T, Rs(s), Rs(o), Rec}$$

$S_5$  – sequence of system operation in case of user existence, successful scanning of space and successful scanning of objects:

$$S_5 = \overbrace{I, L, T, S(s), S(o), Rec}$$

$S_6$  – sequence of system operation in case of user existence, successful scanning of space and unsuccessful scanning of objects:

$$S_6 = \overbrace{I, L, T, S(s), Rs(o), Rec}$$

$S_7$  – sequence of system operation in case of user existence, unsuccessful scanning of space and successful scanning of objects:

$$S_7 = \overbrace{I, L, T, Rs(s), S(o), Rec}$$

$S_8$  – sequence of system operation in case of user existence, unsuccessful scanning of space and unsuccessful scanning of objects:

$$S_8 = \overbrace{I, L, T, Rs(s), Rs(o), Rec}$$

$L_1, L_{11}, L_{12}, L_{13}, L_{14}$  - check for the presence of a user:

$$\begin{array}{l}
 L_{11} = \overline{S_1 ; S_5 ; u_1^-?} \quad L_{12} = \overline{S_2 ; S_6 ; u_1^-?} \\
 L_{13} = \overline{S_3 ; S_7 ; u_1^-?} \quad L_{14} = \overline{S_4 ; S_8 ; u_1^-?} \\
 L_1 = \overline{\left( \begin{array}{l} S_1 \\ S_2 \\ S_3 \\ S_4 \end{array} ; \left( \begin{array}{l} S_5 \\ S_6 \\ S_7 \\ S_8 \end{array} ; u_1^-? \right) \right)}
 \end{array}$$

$L_2, L_{21}, L_{22}, L_{23}, L_{24}$  - check for the success of scanning the workspace:

$$\begin{array}{l}
 L_{21} = \overline{S_1 ; S_3 ; u_2^-?} \quad L_{22} = \overline{S_2 ; S_4 ; u_2^-?} \\
 L_{23} = \overline{S_5 ; S_7 ; u_2^-?} \quad L_{24} = \overline{S_6 ; S_8 ; u_2^-?} \\
 L_2 = \overline{\left( \begin{array}{l} S_1 \\ S_2 \\ S_5 \\ S_6 \end{array} ; \left( \begin{array}{l} S_3 \\ S_4 \\ S_7 \\ S_8 \end{array} ; u_2^-? \right) \right)}
 \end{array}$$

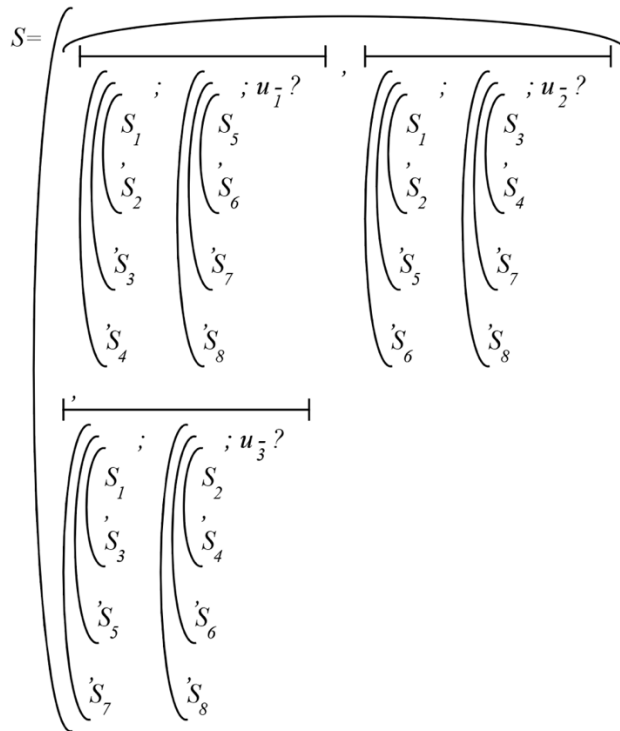
$L_3, L_{31}, L_{32}, L_{33}, L_{34}$  - check for successful scanning of items:

$$\begin{array}{l}
 L_{31} = \overline{S_1 ; S_2 ; u_3^-?} \quad L_{32} = \overline{S_3 ; S_4 ; u_3^-?} \quad L_3 = \overline{\left( \begin{array}{l} S_1 \\ S_3 \\ S_5 \\ S_7 \end{array} ; \left( \begin{array}{l} S_2 \\ S_4 \\ S_6 \\ S_8 \end{array} ; u_3^-? \right) \right)} \\
 L_{33} = \overline{S_5 ; S_6 ; u_3^-?} \quad L_{34} = \overline{S_7 ; S_8 ; u_3^-?}
 \end{array}$$

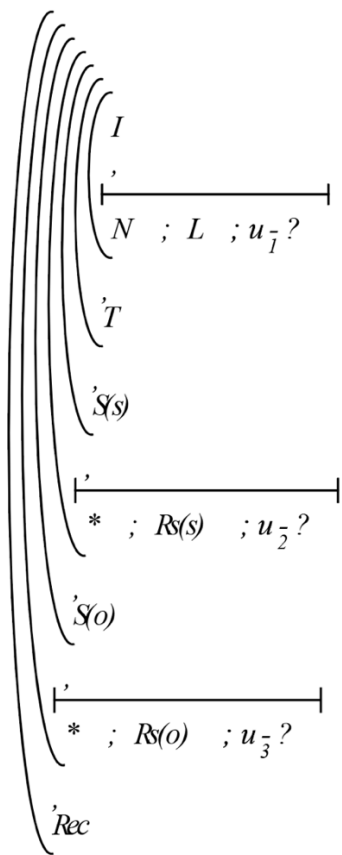
The general formula without substituting nested eliminations will look like this:

$$S = \overbrace{L_1, L_2, L_3}$$

After substituting the eliminations, the general formula will look like this:



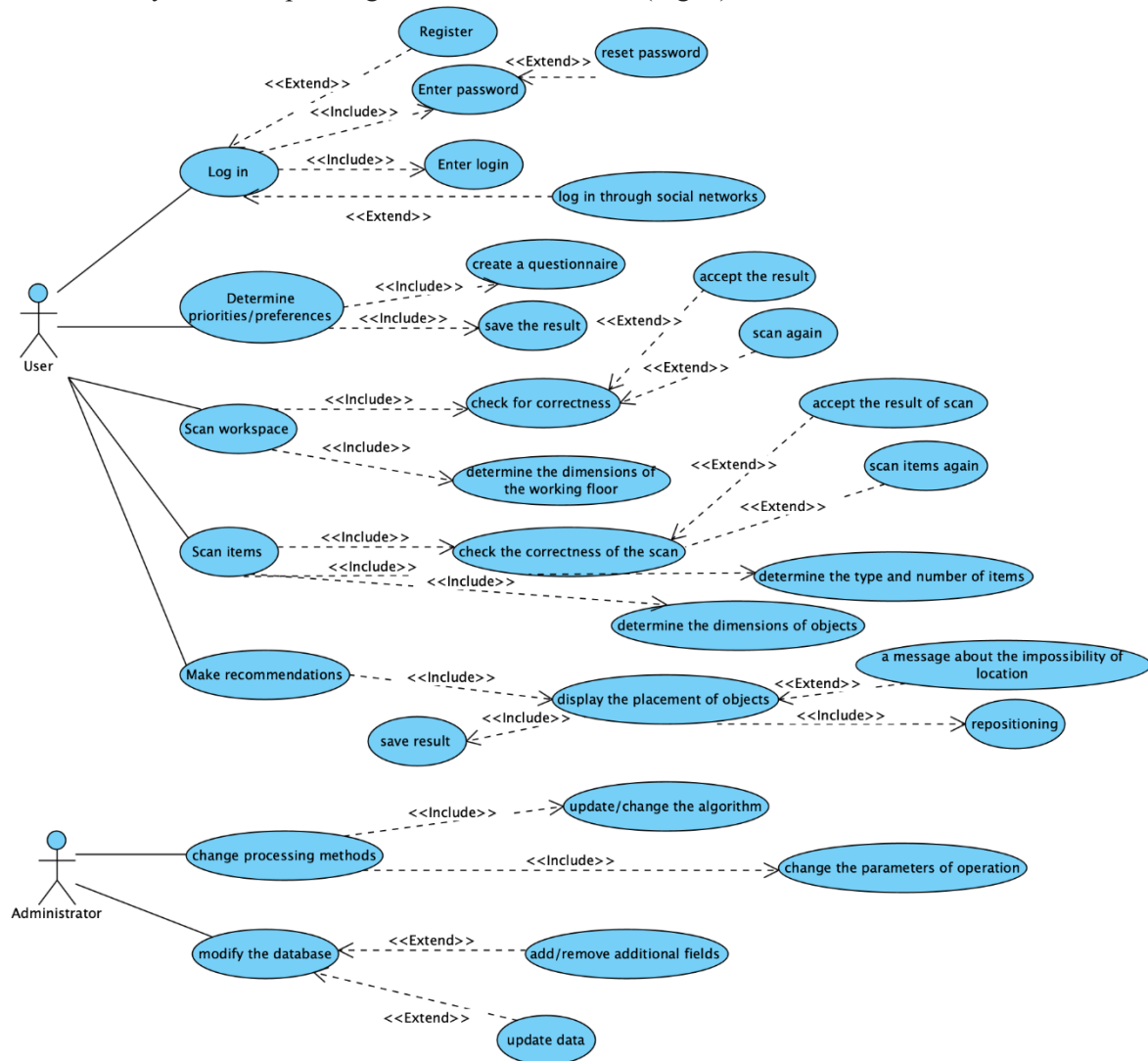
The next stage is the substitution of the corresponding sequences in the elimination. As a result of using the properties of the algebra of algorithms [17], we subtract the common uniterms by the sign of the elimination operation and obtain the following formula of the algebra of algorithms:



The next stage of the research was the design of the system using an object-oriented approach [18, 19]. Namely, at the beginning of the design, a diagram of use cases was created, made in the Visual Paradigm software environment. The use case diagram shows a view of the system that allows:

- define the common boundaries and context of the simulated software;
- formulate general requirements for the functional behavior of the designed system;
- develop the initial conceptual model of the system for its further detailing in the form of logical and physical models;
- prepare source documentation for the interaction of system developers with customers and users.

The diagram defines two main actors - the User and the Administrator, which are connected by a generalization relationship, that is, all the actions that the User can perform, as well as some additional ones defined by the corresponding role, will be available (Fig. 2).



**Figure 2:** Diagram of use cases

So, the main precedents for the User actor are:

- Log in.
- Includes login input - this will allow the user to enter a unique identifier to identify the latter.
- Includes entering a password - this will allow the user to go through the authentication process to access the system.
- Extends the password recovery precedent - will allow the user to recover a lost password.
- Extends login to the system using social network resources - will allow the user to log in through social networks, including Facebook, Linked, etc.
- Determine priorities/preferences – implemented by conducting a short survey to form individual preferences.

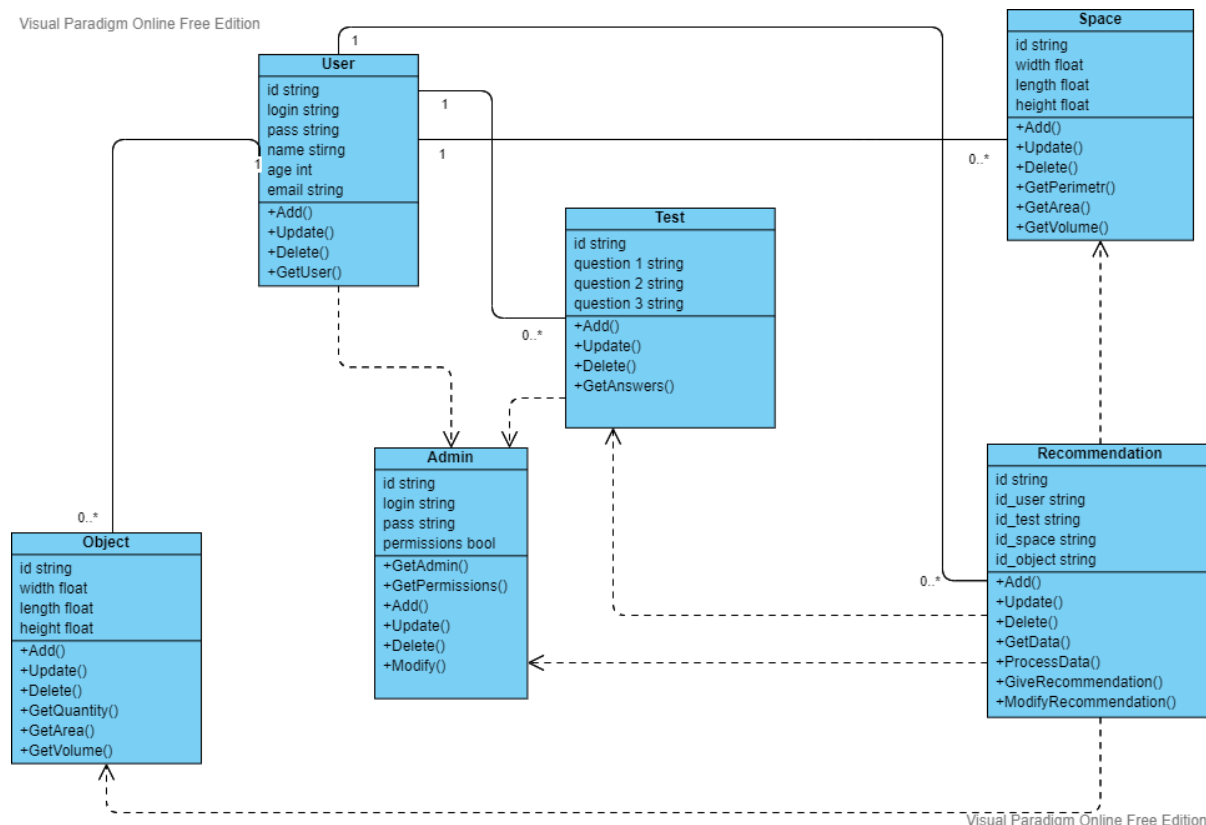


- Scan the workspace - involves taking a photo of the workspace that needs to be organized.
- Includes checking for correctness - checking that the space is photographed correctly and that there are no redundant/deficient elements.
  - Extends Accept the result - if everything is photographed correctly, then the result is saved.
  - Expands Scan again - if the photo is incomplete or contains extra elements, then you need to scan again.
- Includes Determine the size of the working space - after a successful scan, the system determines the size of the space on which it is necessary to organize.
- Scan objects - involves photographing objects that must be additionally placed within the workspace.
  - Includes check for correctness
    - Extends Accept the result - if everything is photographed correctly, then the result is saved.
    - Extends Repeat scanning - if the photo is incorrect, there is an opportunity to repeat this process.
  - Includes Determine Dimensions - determines the size of each item to be placed
  - Includes Determine type and number of objects - performs type detection and counts the number of objects.
  - Create recommendations - after processing all received data, the system displays possible options for organizing the workspace.
  - Enables Show placement of objects - recommendations are presented as placed objects in the scanned space.
    - Expands Unable to locate - if there is no option given the input data, the system issues a message about the impossibility of locating.
    - Enables Relocation - allows you to display a different location recommendation.
  - Includes Save result - save the received results.

The main precedents of the Administrator actor are:

- Change the methods of functioning - will allow the administrator to make changes to the algorithms by which the system works.
  - Expands Update/change the work algorithm - the ability to update the algorithms of the system's operation.
  - Extends Change the parameters of operation - the ability to change certain parameters during the operation of the system.
  - Modify database - allows the administrator to modify the database in which all results will be stored.
    - Expands Add field - the ability to add a new field.
    - Extends Refresh data - the ability to refresh data.

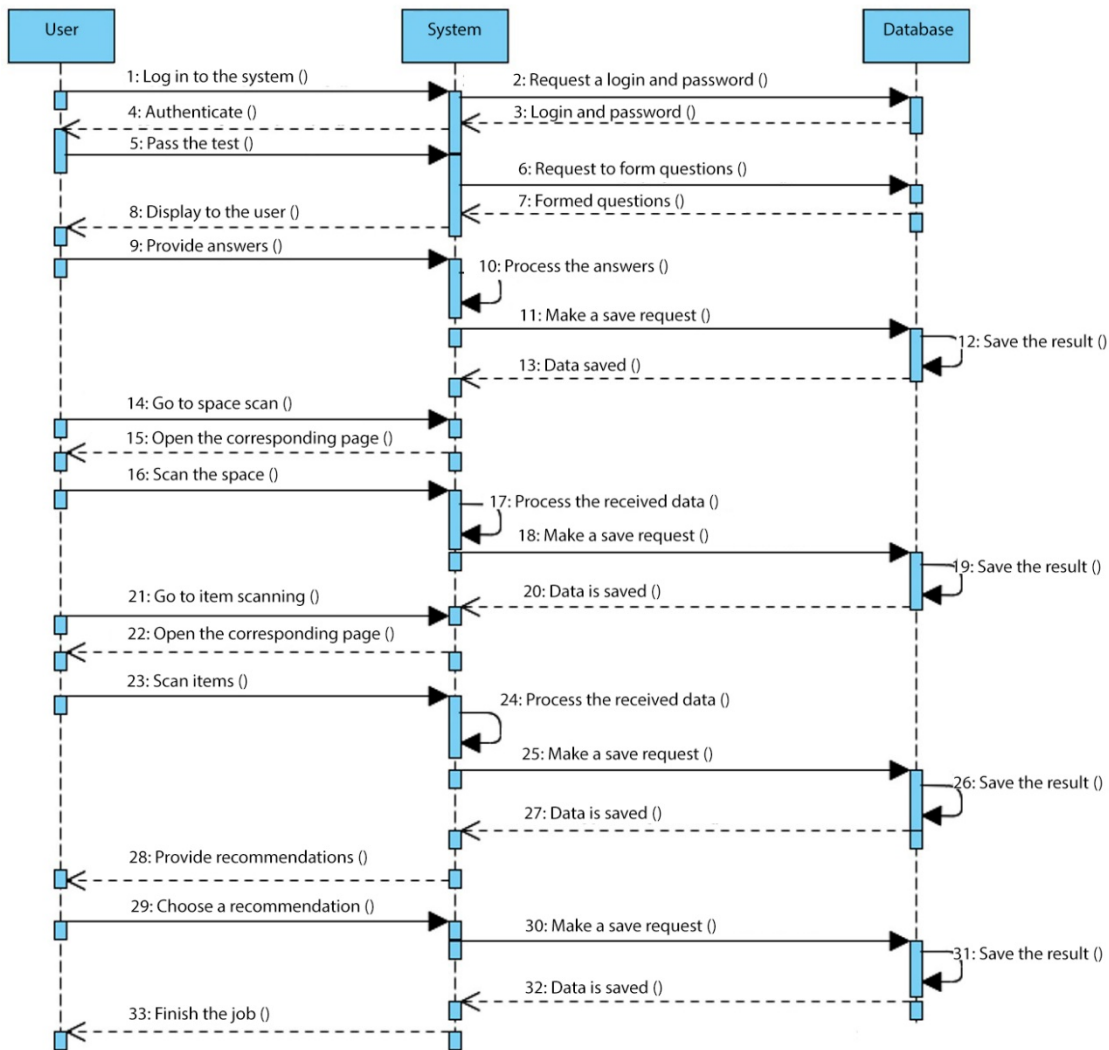
Next, according to the concept of object-oriented design, a class diagram will be constructed. The class diagram is the basis of any software project documentation and without its implementation further correct construction is impossible. It allows you to formalize the logical model of the program at the level of structural elements of the system and provides a static representation of the structure of the program. The class diagram of the designed system is shown in Fig. 3.



**Figure 3:** Class diagram

At the initial stage of development, 6 classes were created: User, Space, Object, Test, Admin, Recommendation. The User class contains the following attributes – id string, login string, pass string, name string, age int, email string and the following operations – Add(), Update(), Delete(), GetUser(). Designed to describe information about the user. The Test class contains the following attributes – id string, question 1 string, question 2 string, question 3 string and the following operations – Add(), Update(), Delete(), GetAnswers(). It is intended for testing to determine user priorities/preferences. The Space class contains the following attributes – id string, width float, length float, height float and the following operations – Add(), Update(), Delete(), GetPerimetr(), GetArea(), GetVolume. Intended for displaying workspace parameters. The Object class contains the following attributes – id string, width float, length float, height float and the following operations – Add(), Update(), Delete(), GetQuantity(), GetArea(), GetVolume. Defines the characteristics of individual objects that must be located on the plane. The Recommendation class contains the following attributes – id string, id\_user string, id\_test string, id\_space string, id\_object string and the following operations – Add(), Update(), Delete(), GetData(), ProcessData(), GiveRecommendation(), ModifyRecommendation( ). It contains recommendations that will be generated for the user. The Admin class contains the following attributes id string, login string, pass string, permissions bool and the following operations GetAdmin(), GetPermissions(), Add(), Update(), Delete(), Modify(). It contains a description of the features of the administrator's interaction.

In order to display the features of the interaction between individual objects, a sequence diagram was constructed. This type of diagram is one of the ways to formalize usage scenarios, and its advantage is that at the initial stages of describing the scenarios, it is possible to find out the composition of interacting components and describe the flow of messages between them. The sequence diagram for the designed system is shown in Fig. 4. It shows three system objects - User, System and Database. After logging in to the User's system, the User begins to go through the main flow in the system - passing the test, scanning the space, scanning objects and receiving recommendations.



**Figure 4:** Sequence diagram

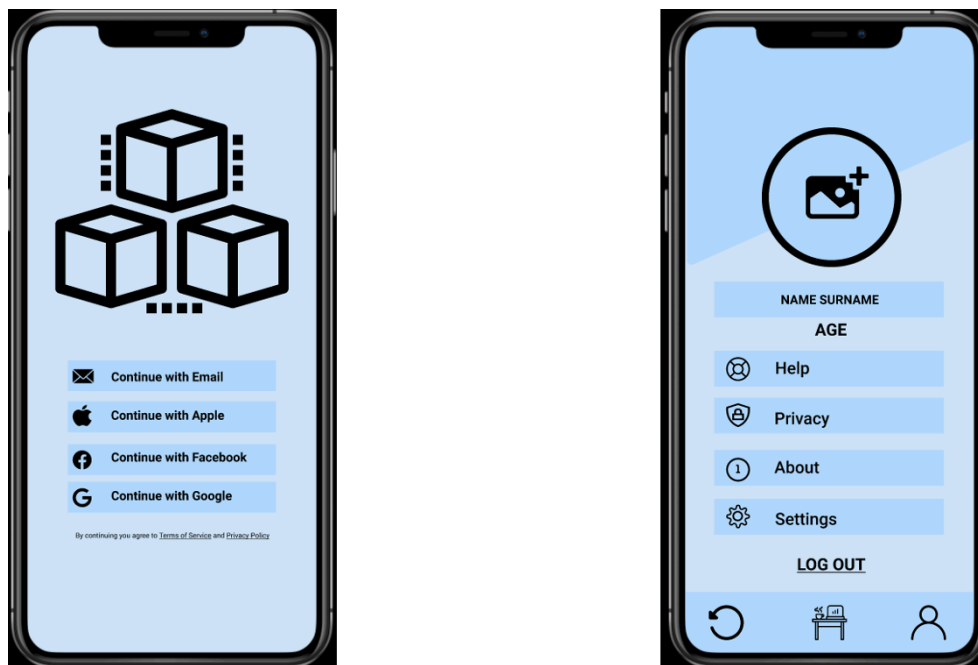
The development of any system cannot be complete without a choice of methods for solving the task. After analyzing the literary sources, a decision was made to use the pattern recognition method in tandem with the recommendation engine to cover the user's needs. The basis of the developed system is a recommendation engine based on the content-based filtering method [20]. This method is chosen because such a model does not require any extraneous data, as the recommendations are specific to a particular user. At the same time, the model can cover the specific interests of the user and recommend niche items that interest him. Considering the recommended type of system, the main stages of the latter are:

- the information gathering phase, which involves gathering user information to create a user profile/model for predictive tasks, including user attributes, behavior, or content of resources accessed by the user. In the process of collecting information, we will use the technology of explicit feedback, which involves providing ratings for preferences through the system interface, in order to build and improve the model.
- learning phase, a learning algorithm is applied to filter and use user features based on the feedback collected in the information gathering phase.
- prediction/recommendation phase – recommends which items the user might prefer. This can be done either directly from the data set collected in the information gathering phase, which can be memory-based or model-based, or through the observed actions of the system through the user.

Since the organization of personal space depends on a number of factors, subjective factors are also planned to be taken into account in the process of forming tests/recommendations. In particular, according to studies [21], it can be seen that people with a more mathematical mindset like everything to be organized, on shelves and clearly. At the same time, creative people prefer "conditional chaos". Therefore, in order to provide better recommendations in the system, it is planned to use the studies of Roger W. Sperry [22], which allow including in the test the question of determining the hemisphere that is dominant in an individual user. Namely, studies have shown that people dominated by the left part are usually analytically oriented, and the right, respectively, are more creative [23]. Therefore, determining whether a user perceives information better with the right hemisphere or with the left directly depends on how exactly he needs to arrange objects on his desktop.

The next stage was the design of the system, which, based on the analysis, was decided to be implemented in the form of a client-server system, while the client part was implemented in the form of a mobile application running under the control of the Android operating system. As an integrated development environment, Android Studio [24, 25], specially developed by Google Corporation for the Android operating system, is used. The MySQL software [26,27], which provides support for distributed applications and is included in most cloud platforms, was chosen as the database management system. The development architecture is based on the Rest API technology, which is a server framework of a mobile application that offers significant scalability necessary for the pattern recognition process [28]. In order to integrate the pattern recognition method into the system, ready-made Python libraries were used, namely, Crab is a flexible, fast recommendation engine for Python, which integrates classic recommendation algorithms and information filtering into ready-made Numpy, Scipy, and Matplotlib packages [29, 30]. The developed prototype of the application is presented in fig. 5-8.

Registration in the application is available in four ways: through email, Apple (if the application for IOS is expected in the future), Facebook and Google account [31]. To display a typical work scenario, the "Continue with Email" option is selected. After selection, you will be transferred to the next page, where you need to enter your email and registration data: First name, last name, age and password. After successful entry, click the "Sign up" button, as a result, the system will redirect us to the created profile.



**Figure 5:** Home page and created profile

After clicking on the middle icon, you can proceed to the first step, which is to pass the test. After completion of which you can see a screen where the system congratulates you for passing and there are two options: take the test again by clicking "Take test again" or continue by clicking "Next" (Fig. 6.)

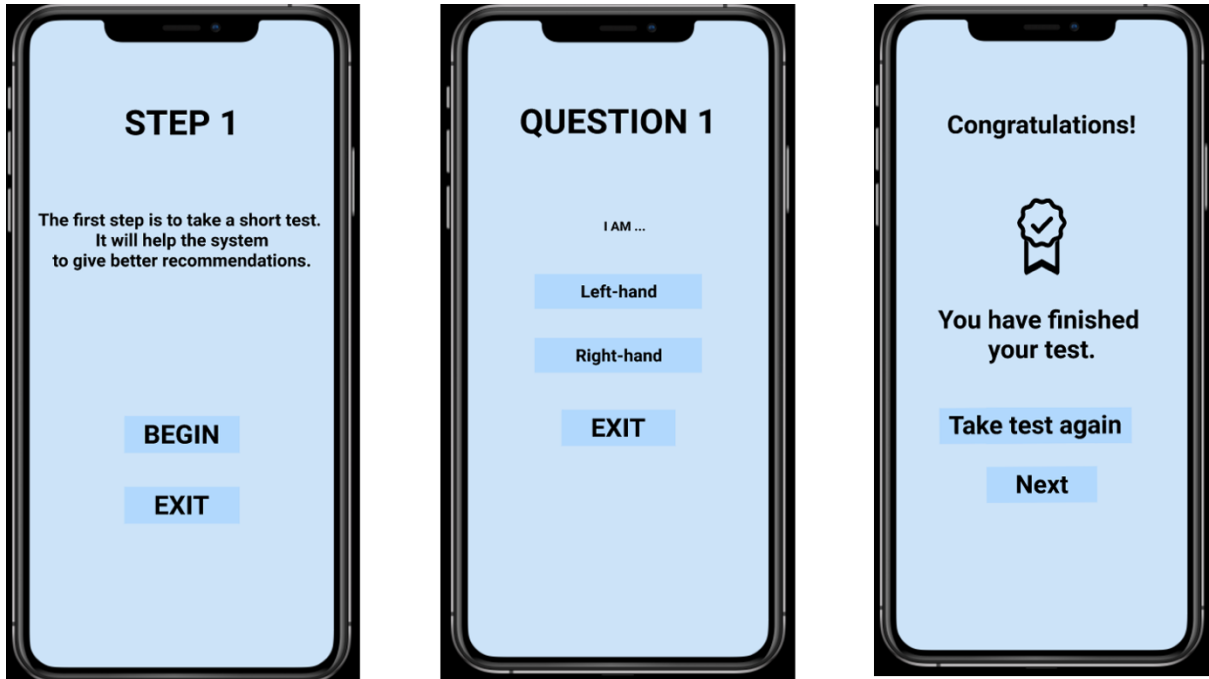


Figure 6: Testing

After clicking the "Next" button, the user goes to the next step, namely: space scanning. After clicking on "Let's scan", the system goes into scanning mode, covering the area of space. After selecting the "Next" button, the user goes to the third step, namely: scanning objects. After scanning, there are two buttons: Save - save the item and Retake - repeat the scan. In this way, element-by-element scanning of all objects is carried out. (Fig. 7)

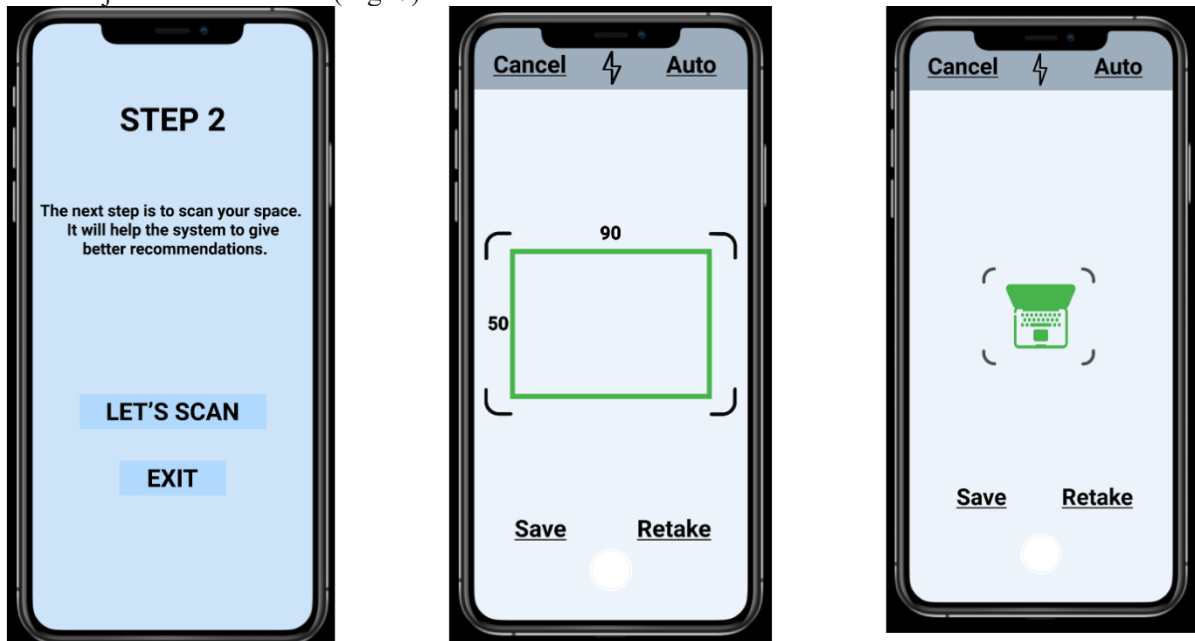


Figure 7: Scanning process

Next, the entered data is processed. After all the data has been processed, the system displays possible options for placing objects in space (Fig. 8).

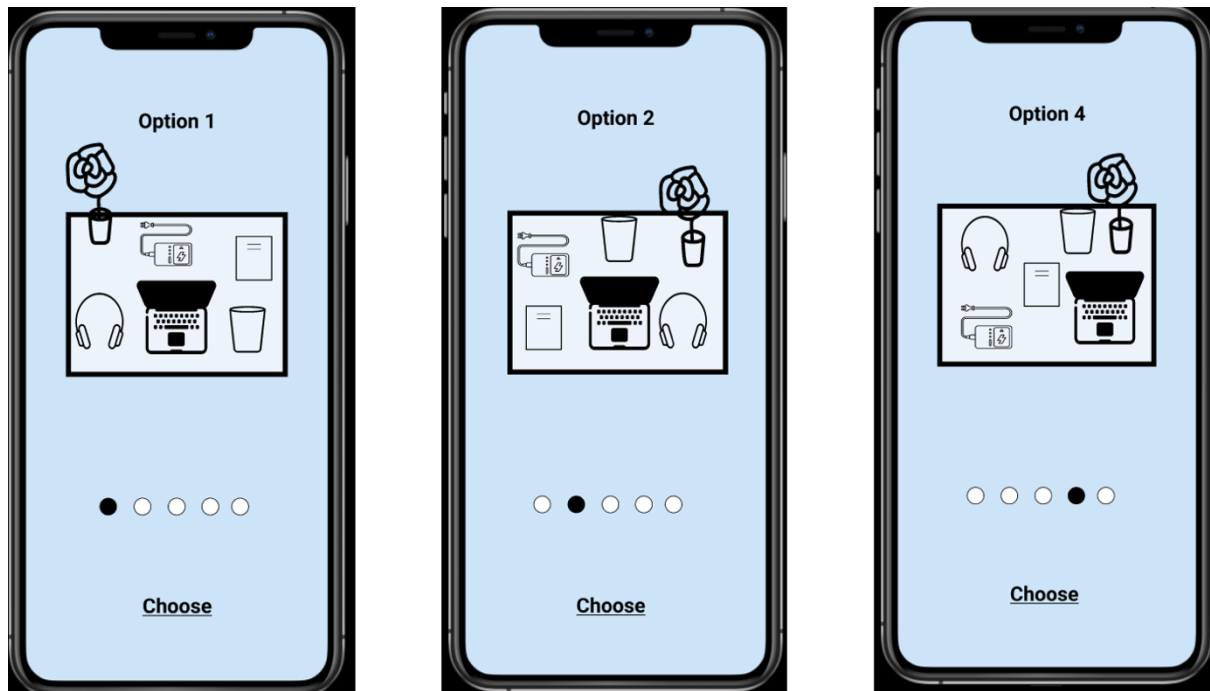


Figure 8: Formed options for elements placement

### 3. Conclusion

As a result of the conducted research, the existing methods and known systems that provide tools for organizing the workspace were analyzed. An analysis of the subject area was carried out, and it was shown that today there is a limited set of tools (libraries) that are used in the process of pattern recognition, among the most well-known are: DeepFace, TensorFlow and OpenCV, and therefore the lack of necessary tools makes the task of designing and implementing a work order system urgent. The next stage of the work was a systematic analysis of the subject area with the formation of dissection goals and their presentation in the form of a goal tree. Next, a mathematical description of the subject area was given using the apparatus of the algebra of algorithms, which provided the necessary background for the formation of mathematical support. Object-oriented design of the software system was carried out by constructing a set of diagrams in UML notation, some of which (use case diagrams, classes and sequences) are presented in the work. Based on the conducted research, the system was designed and implemented in Android Studio. The created software product works in prototype mode and implements the described functionality.

Further research will be directed to the verification and approval of the obtained results and expansion of the functionality.

### 4. References

- [1] C. Blakeman, *Rehumanizing The Workplace By Giving Everybody Their Brain Back*. Crankset Group Publishing, 2020
- [2] M. Thompson, *Working Space: An Insight into the Creative Heart*. Hardie Grant Books, 2013
- [3] D. Charnas, *Work Clean: The life-changing power of mise-en-place to organize your life, work, and mind*. Rodale Books, 2016
- [4] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006
- [5] U. Braga-Neto, *Fundamentals of Pattern Recognition and Machine Learning*. Springer, 1st ed., 2020
- [6] DeepFace – The Most Popular Open Source Facial Recognition Library, URL. <https://viso.ai/computer-vision/deepface/>

- [7] A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, 2019
- [8] D. Storti, CUDA for Engineers: An Introduction to High-Performance Parallel Computing. Addison-Wesley Professional, 2015
- [9] J. Howse, J. Minichino, Learning OpenCV 4 Computer Vision with Python 3: Get to grips with tools, techniques, and algorithms for computer vision and machine learning, Packt Publishing, 2020
- [10] C. Huyen, Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications, O'Reilly Media; 1st edition, 2022
- [11] N. Shakhovska, N. Boyko, P. Pukach, The information model of cloud data warehouses // Advances in Intelligent Systems and Computing (AISC). – 2019. – Vol. 871: Advances in intelligent systems and computing III. Selected papers from the International conference on computer science and information technologies, CSIT 2018, September 11-14, Lviv, Ukraine. – P. 182–191
- [12] O. Mediakov, T. Basyuk, Specifics of Designing and Construction of the System for Deep Neural Networks Generation // CEUR Workshop Proceedings. – 2022. – Vol. 3171 : Computational Linguistics and Intelligent Systems 2022 : Proceedings of the 6th International conference on computational linguistics and intelligent systems (COLINS 2022). Vol. 1 : Main conference, Gliwice, Poland, May 12-13, 2022. – P. 1282–1296.
- [13] G. Blokdyk, Collaborative filtering Second Edition. 5STARCOoks, 2021
- [14] T. Basyuk, A. Vasyliuk, V. Lytvyn, Mathematical Model of Semantic Search and Search Optimization, Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Systems (COLINS-2019). Volume I: Main Conference, Kharkiv, Ukraine, April 18-19, 2019, Vol-2362: pp.96-105.
- [15] O. Veres, P. Ilchuk, O. Kots, Data Science Methods in Project Financing Involvement, Proceedings of the 16rd International Conference on Computer Sciences and Information Technologies (CSIT-2021). Volume 2, pp. 411-414.
- [16] A. Dennis, B. Wixom, D. Tegarden, Systems Analysis and Design: An Object-Oriented Approach with UML. Wiley, 2015.
- [17] V. Ovsyak, ALGORYTMY: metody pobudovy, optymizatsiyi, doslidzhennya virohidnosti. – Svit, 2001.
- [18] D. Pilone. UML 2.0 Pocket Reference: UML Syntax and Usage, O'Reilly Media, 2006.
- [19] A. Vasyliuk, T. Basyuk, Construction Features of the Industrial Environment Control System, Proceedings of the 5rd International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021). Volume I: Main Conference, Kharkiv, Ukraine, April 22-23, 2021, Vol-2870: pp.1011-1025.
- [20] C. Charu, Aggarwal Recommender Systems: The Textbook, Springer, 2016
- [21] P. Barber, D. Legge, Perception and Information. Routledge, 2019
- [22] M. Gazzaniga, Tales from Both Sides of the Brain: A Life in Neuroscience. Ecco, 2015
- [23] N. Hammond, Left Brain vs. Right Brain: What Does This Mean for Me? URL. <https://www.healthline.com/health/left-brain-vs-right-brain>
- [24] B. Zapata, Android Studio 2 Essentials. Springer, 2016
- [25] S. Uzayr, Mastering Android Studio. CRC Press, 2022
- [26] V. Grippa, S. Kuzmichev, Learning MySQL: Get a Handle on Your Data. O'Reilly Media, 2021
- [27] D. Nichter, Efficient MySQL Performance: Best Practices and Techniques. O'Reilly Media, 2022
- [28] M. Masse, REST API Design Rulebook. O'Reilly Media, 2011
- [29] A. Agresti, M. Kateri, Chapman Foundations of Statistics for Data Scientists: With R and Python. Chapman & Hall/CRC Texts in Statistical Science, 2021
- [30] W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media, 2017
- [31] E. Burnette, Hello, Android: Introducing Google's Mobile Development Platform. Pragmatic Bookshelf, 2015