DOI: 10.20998/2079-0023.2023.02.07
UDC 004.9

V. V. YAMBURENKO, National Technical University "Kharkiv Polytechnic Institute", Student, Kharkiv, Ukraine, e-mail: yamburenko@gmail.com, ORCID: https://orcid.org/0009-0005-4797-8684
I. V. LIUTENKO, Candidate of Technical Sciences (PhD), Docent, National Technical University "Kharkiv Polytechnic Institute", Associate Professor at the Department of Software Engineering and Management Intelligent Technologies, Kharkiv, Ukraine, e-mail: cherliv68@gmail.com, ORCID: https://orcid.org/0000-0003-4357-1825
A. M. KOPP, Doctor of Philosophy (PhD), Docent, National Technical University "Kharkiv Polytechnic Institute", Associate Professor at the Department of Software Engineering and Management Intelligent Technologies, Kharkiv, Ukraine, e-mail: kopp93@gmail.com, ORCID: https://orcid.org/0000-0002-3189-5623
D. M. HOLOVNIA, National Technical University "Kharkiv Polytechnic Institute", Student, Kharkiv, Ukraine, e-mail: dimagolovnya@gmail.com, ORCID: https://orcid.org/0009-0009-0901-025X

ALGORITHM AND SOFTWARE OF MEDICAL PERSONNEL SELECTION SYSTEM

There is a lot of routine work in any organization, including in recruitment agencies. Effective management organization and automation of activities of employees of recruiting agencies is not an easy task. The system should automate the routine actions of workers of recruiting agencies and be convenient for their clients. This paper proposes an approach to automating the selection of necessary medical staff. Not all information systems used by recruiting agencies can compare candidates and generate results that include several of the best candidates. Based on the analysis of the subject area, groups of parameters that significantly affect the choice of medical personnel were determined. The proposed approach is to analyze the request from the client, and then in the system find requests of other clients similar to it in terms of parameters, for which a candidate has already been found. The next step is to take the profiles of healthcare professionals that have been suggested for these requests (they act as benchmarks) to further compare them with existing candidates. Each employee profile parameter has its own similarity function. Available candidates will receive scores and will be ranked. We also additionally adjust the assessment by comparing candidates with the current request. Software was developed to automate the selection of medical personnel. For its implementation, a three-level client-server architecture is proposed. MVC (Model View Controller) architecture was chosen for the server part. The Single Page Application architectural template is used for the client part. The server part is divided into three layers, which further demarcate and structure the responsibilities of the system components. .NET technologies are used to implement business logic. SQL Server is used for the server and database provider. The use of the software implementation of the developed system demonstrated quite good results. The average time for selecting the 10 best candidates out of 500 is 0.4 seconds, and the processing of only 1 resume by a person takes several minutes.

Keywords: personnel selection, automation, data processing, algorithm, software, system architecture, development technologies.

Introduction. In the era of information overload, people use a variety of strategies to make many decisions: what product and service to choose, how to spend your free time, what movie to watch, who to meet, where to repair your car, or which candidate is more suitable for a particular job. To speed up decision-making and simplify the process, information systems that automate some of these strategies come to the rescue in order to provide individual, accessible and high-quality recommendations for further choices. The use of information systems in the field of medical personnel selection can be extremely relevant for several reasons.

First, medical personnel play a crucial role in providing quality medical care, so it is important to find the most competent specialists to work in medical facilities. These systems can analyze volumes of data about potential candidates, take into account their skills, education, work experience, as well as specialization and other factors that influence successful performance in the medical environment.

Second, information can effectively reduce the time spent by recruiters on searching for and evaluating candidates, which increases productivity and lowers costs for the recruitment process. In addition, such systems can provide a more objective approach to candidate selection, as they are based on data analysis rather than subjective assessments.

The purpose of the work is the development of algorithmic and software for the task of automation of personnel selection in the field of medicine. This will improve the quality of the selection of doctors and reduce the amount of time needed to find the right specialist.

Analysis of recent research and publications. Recommendation systems are very often used to solve selection problems [1]. The field of data analysis is constantly developing and improving, and the direction of recommender systems is one of the most promising in it. Today, we can find a huge number of examples of the use of recommender systems in many areas of our life [2], [3]. As the simplest example, in advertising, for different users, the ad is always tailored to them. Popular services, such as Google Ads, Yahoo Gemini, Microsoft Advertising, create hidden profiles for users, and use them to personalize the search results and add personal advertising positions to it. The same approach is used in popular social networks such as Twitter, Facebook, and Instagram, where we see personalized advertising between posts.

The first commercially available recommendation system was developed by Amazon and was called Recommendations for You. This system was implemented in 1998 [4]. The goal was to improve the personal experience of customers and increase sales. The PC analyzed each user's purchases and product views, and then recommended other products they might like based on that information.

Technically, recommender systems have a lot in common with forecasting theory, information retrieval systems, approximation theory and the science of

management and modeling of consumer choice, but modern recommender systems are considered as an independent scientific field. The main task of the recommender system is to determine ratings for objects, which can be goods and services or even other users, which are not yet known to a specific user or consumer [5]. These ratings are based on information about the customer's previous transactions or other characteristics related to his preferences, actions or personal information [6]. The system analyzes the data, assigns a rating to the objects and chooses those with the highest ratings to offer them to the user [7].

In the medical field, recommendation systems are used in several directions. The first is decision-making, when it is necessary to analyze a certain case and make a certain decision related to treatment. The second is the selection of medical personnel, when the patient is looking for, for example, a good family doctor and the system tries to recommend someone to him. Or when a certain organization, such as a rehabilitation center or a home for the elderly, is looking for a certain specialist with the necessary skills to solve certain tasks (providing therapies, care). Usually, such organizations turn to certain recruiting centers that search for and select potential candidates. Such recruiting centers can use recommender systems for an optimal and quick search for specialists to close a specific request.

HealthNetsocial [8] is an Italian social network in alpha testing. This network connects patients with similar health situations. This system helps find solutions to health problems through recommendations from doctors or medical organizations that best match the user's profile. The key part of the system is the recommender system, which suggests other patients who are most similar to the user and helps with the choice of a doctor or hospital according to the state of health. Their algorithm first calculates the similarity between patient profiles. And then it generates a ranked list of doctors or medical institutions that best match the patient's profile. In addition, data on health conditions and previous cases of other patients are also taken into account to clarify the results [5].

The University of California and the Nova School of Business and Economics have jointly developed a prototype algorithm [9] that helps match primary care physicians to patients. This research was sponsored by grants from Amazon and Microsoft. For their study, they used a large dataset from a private European provider of health advice. Which contained 1.07 million records of consultations between 380 thousand patients and 314 doctors in 16 hospitals for the period 2012-2017. As a method of solving the generation problem, they used the method of collaborative filtering. Their task was to generate a rating of patients' trust in doctors, that is, as a result, the user will receive a list of doctors whom he is likely to trust the most [6].

In 2019, the article "An adaptive doctor-recommender system" was published, which considers the research process of the problem of generating recommendations for the selection of doctors [10]. The study was conducted by several universities in Pakistan (NUML, IIU UET). A hybrid recommendation system was proposed and implemented, combining content, collaborative and demographic filtering approaches for efficient generation of a doctor's recommendation. The proposed system solves the problem by analyzing the patient's interest in choosing a doctor. And also uses a relatively new adaptive algorithm for generating the rating of doctors. Moreover, the rating function converts patients' criteria for choosing doctors into a numerical value that will be used in the rating of doctors. The main feature of the system is the use of an analytically hierarchical model, which was presented by Thomas Saati.

Formulation of the problem. In general, the use of information systems in the field of medical personnel selection can improve the quality of recruitment, ensure the optimal distribution of staff and contribute to the growth of the efficiency of medical institutions. We need to develop software that, based on certain information and characteristics of medical professionals and information about users looking for a specific specialist, will be able to provide quality recommendations for the selection of medical professionals.

The following information about the needs and requirements of the user (details of specialization, personal characteristics, information about experience) should be submitted to the input of such a system, which will help to build a basic portrait of the necessary specialist and user, and information about the user's past operations with the system, namely, with which specialists he has already interacted and how satisfied he was with it (quality assessment from the user).

At the output, the system should generate ratings for possible recommended specialists, select those with the highest ratings and offer them to the user.

Proposed solutions. During the analysis of the subject area, the following parameters were determined that can be used when solving the task of generating recommendations. Thus, the parameters can be conditionally divided into three groups, depending on which entity they belong to.

The first group corresponds to the medical worker and the vacancies posted by the client, it covers the main parameters that recruiters take into account when searching for candidates. These parameters are:

- age;
- sex;
- education and its degrees;
- professional experience and its duration;
- availability of certifications, licenses, medical certificates, vaccinations;
- list of specializations;
- expected salary;
- language knowledge.

The second group refers to the client itself (a company looking for an employee), namely the minimum necessary information about the client that can be used to clarify the assessment. In it, the parameter is information about past requests.

The third group takes into account parameters corresponding to past inquiries and contains certain information that can be used to analyze the success of the

recruitment. Such parameters are: the status of the request (hiring took place or not); employee assessment; reviews

The key entities in the system are the profiles of medical workers, system clients, requests for the search of medical workers. The point of searching for a candidate is always a search request that has a certain set of characteristics-desires of the client, this is a kind of documented model of the client's preferences. The main idea is to analyze the request from the client, and then find in the system similar requests from other clients that have been successfully implemented (the candidate was found and evaluated by the client).

After finding an array of similar queries, they are ranked by comparison with the current query, where the parameters of both queries are compared to each other to find the level of similarity. The comparison function for each parameter can be arbitrary and depends on the context, but the result will always be a certain number of points that are added to or subtracted from the total similarity value. All requests start with 0 points. In the results, the 10 requests with the highest score, which are considered to be the most similar to the current one, are selected.

The next step is taking the profiles of healthcare professionals that have been proposed and matching to close these ten similar requests to further compare them with existing candidates. That is, further we compare each available free candidate from the system with these ten benchmarks. The principle of comparison is similar to the principles of query comparison. Each employee profile parameter has its own similarity function, which also calculates points. In this way, we rank the available candidates based on their similarity to the benchmarks.

But that's not all, we additionally adjust the assessment by comparing candidates with the current request. Since some parameters of the candidate's profile and the search request are the same, we have the opportunity to compare the candidate's suitability to the requirements of the request. The result of this adjustment increases or decreases the score of each candidate.

A schematic diagram of the algorithm for generating recommendations can be seen in fig. 1.

As a result, after completing three stages of candidate selection, the system should return ten candidates with the highest score. The user will know the number of points

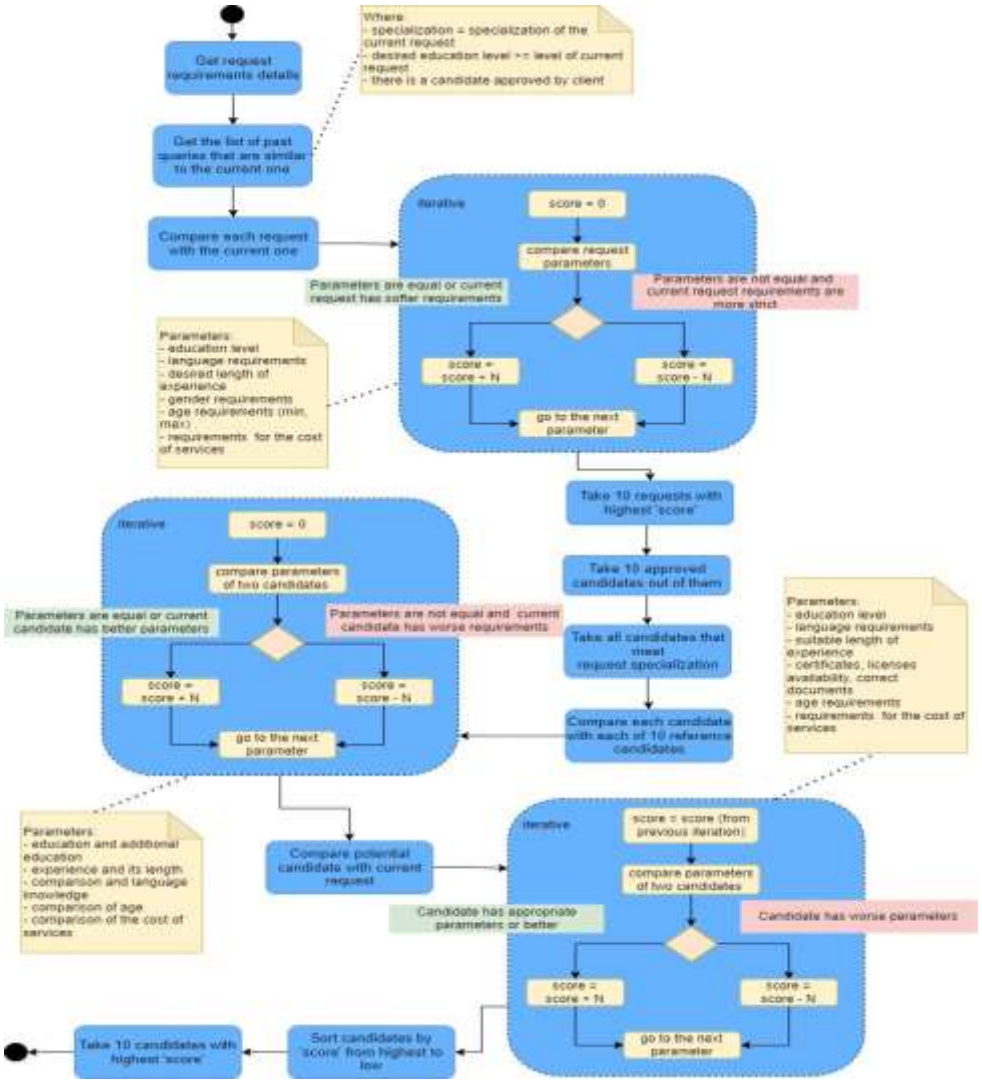

Fig. 1. Scheme of the algorithm for selecting candidates

scored by each candidate, and the candidates will be offered in order from the highest score to the lowest.

To create software, it is necessary to choose a system architecture. Choosing the correct system architecture for implementing a web application is an important step that can significantly affect its quality, performance and scalability. A list of factors that can affect the choice of system architecture is given in [11].

For our solution, it is proposed to choose a three-level client-server architecture, which will allow to remove some risks there.

In our case, data storage will be on a separate database server, which will improve stability and data protection. On another server, there will be a PC that will process the logic. And the user will work directly with a thin client. Fig. 2 shows the general scheme of the chosen three-level client-server architecture.

The client is a Single Page Application, an architectural template that allows you to develop a user interface that never reloads the page, but only dynamically changes its contents. This solution can be developed using the ReactJS library and Redux, which in turn form the React Redux Architecture architectural template. This is a certain analogue of a system based on events and reactions to them.

For the server part, the MVC (Model View Controller) architecture was chosen [12], which allows structuring and dividing the responsibility of individual parts of the server part.

The server part is a set of APIs (Application Programming Interface) for accessing business logic and data. This API is implemented according to the REST (Representational State Transfer) architectural style [13]. It is a set of principles and constraints for building network applications. Web services that are “RESTful” typically use standard HTTP methods (GET, POST, PUT, DELETE, etc.) to perform operations on resources. In the RESTful architecture, "resource" is a key concept, and each resource is identified by a URI (Uniform Resource Identifier). Data in RESTful services is often transmitted in JSON or XML formats.

During the implementation of the software system, a detailed analysis and planning was carried out. The purpose of the analysis was to determine the optimal structure of application components for further convenient maintenance and scaling. These components are a logical continuation of the selected architectures, and only in a more granular way they divide responsibility between system components. To begin with, it is worth considering the component diagram for each application. Fig. 3 shows the general component diagram for the client application.

As we can see, the server part is divided into three layers (fig. 4). These layers further demarcate and structure the responsibilities of system components.

The first layer is the API Layer, its responsibility is to control incoming requests and respond to them. This layer is further divided into groups of components.

API Controllers are direct entry points to the system. Requests from the client system are received here. Controllers clearly understand what to do for each request, which services to call, which error messages to return, which format of the return response. In turn, they use two other groups of components.

Validators are special components that contain all the necessary rules that check the completeness and appropriateness of the input data that comes to the controller.

API Models are a group of components that describe the structure of input and output data.

The next layer is the business logic layer (Business Layer). This layer contains two groups of components.

Services are special components that contain all business logic according to their domain. Each domain entity has its own service. These services carry out data transformation and processing, calculations, etc.

DTO Models are special components – models, which are an alternative representation of domain models, and are used as a means of communication between the API level and the logic level. Regardless, we would have to provide dependencies of the Data Access Layer to the API layer. And this, in turn, would affect the possibility of easily replacing the domain level with another implementation.

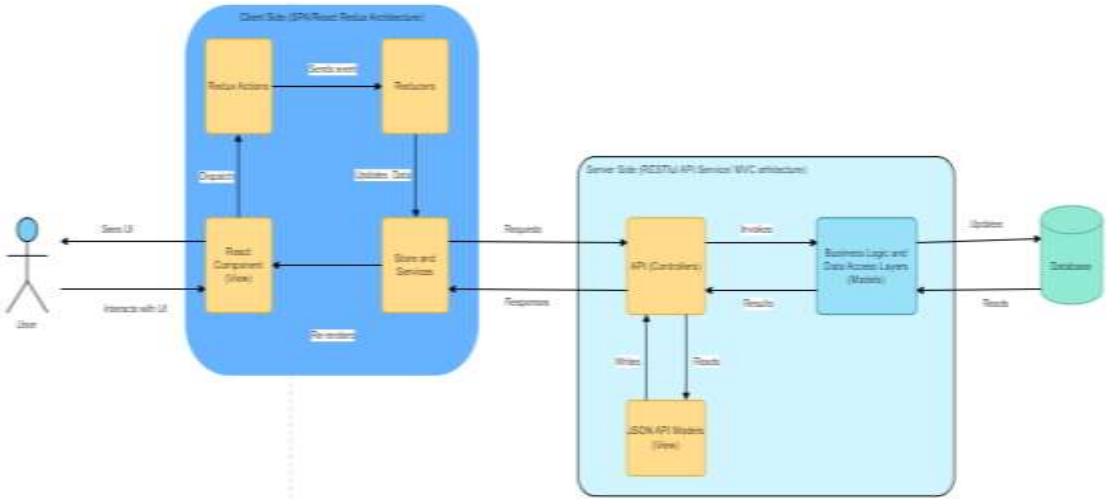

Fig. 2. General scheme of the system architecture

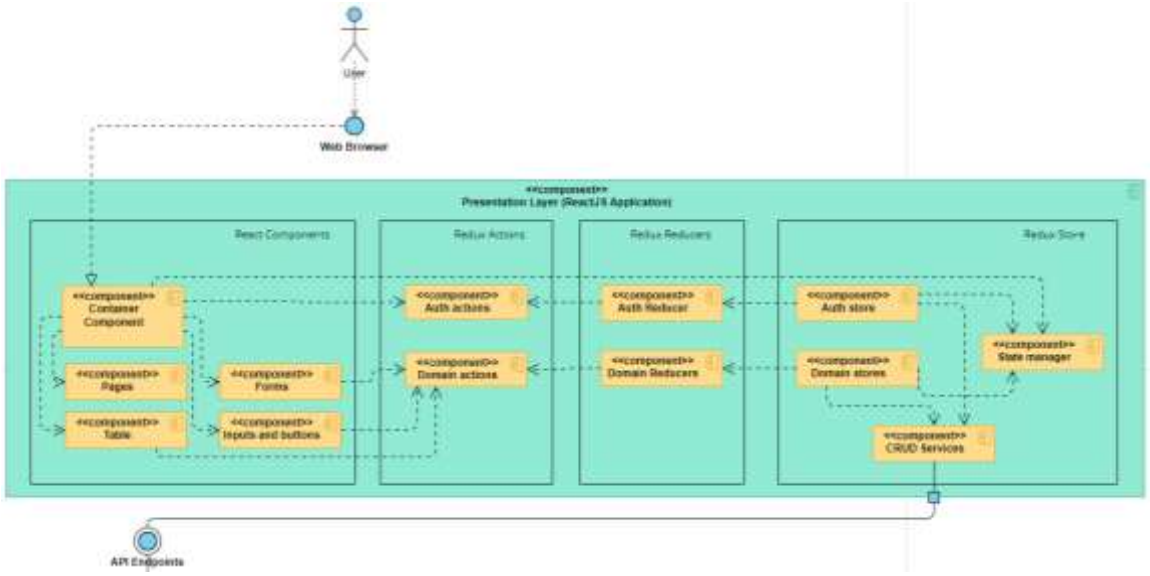

Fig. 3. Diagram of client application components

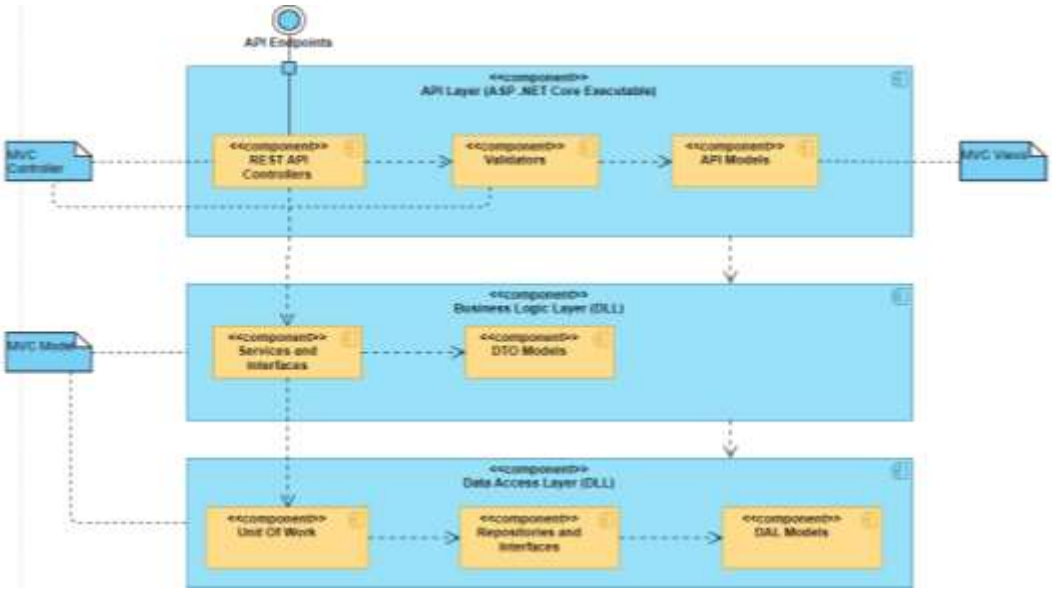

Fig. 4. Diagram of server application components

The last layer is the level of domain models and the level of access to the database, Data Access Layer. This level contains three main components.

UnitOfWork component is responsible for transaction control in the system and access to repositories.

Repositories are a group of components for each entity that contains a group of methods for accessing the database to perform read, write, edit, and delete operations.

DAL Models are domain models, database entity models. They represent the structure for each entity from the database and show the relationships between them.

During the software design work, several class diagrams were developed that reflect the general hierarchy of objects, their relationship types, and attributes. The main domain classes include:

- ClientCompany is a client-company model;
- Candidate is a candidate (medical worker) model;
- HiringRequest is a candidate search request model;
- Discipline is a model of specialization in the medical field;
- Language is a language model that the candidate can sign;
- CandidateEducation is a model of information about the candidate's education;
- WorkExperience is a model of the candidate's work experience;
- User is a system user model required for authorization.

The use of .NET technology was chosen for the server part and business logic. SQL (Structured Query Language) Server was used as a server and database provider. The development environment Visual Studio Community edition was used for the development of the server part. The ReactJS library was used for the client interface.

As a result of the development of the software, all the necessary functionality, including the user interface, was fully implemented.

Experiments were conducted to test the operation of the information system, during which 50 requests were processed and the data of 500 candidates that could be recommended were processed. During the testing, information was collected on the speed of recommendation generation, and proposed candidates were checked for compliance with the vacancy and its requirements. In the table 1 shows several examples of queries that were used for testing.

Table 1 – Examples of candidate search request

Request parameter	Request 1	Request 2	Request 3
Position	Nurse	Surgeon	Trichologist
Direction	Surgery	Neurosurgery	Dermatology
Preferred gender	Man	Is missing	Is missing
Minimum age	30	45	31
Maximum age	55	66	60
Min. rate per hour	59	87	96
Max. rate per hour	151	127	189
Desired level of education	Basic higher education	Master's degree	Master's degree
Duration of experience	5	8	6
Language knowledge	German, Italian, Portuguese	French, Polish	Ukrainian, Spanish

The results of generating recommendations for Request 1: the total number of candidates in the system is 500; the total number of candidates in the system with the chosen specialization is 37; generation time – 0.6 seconds.

The results of generating recommendations for Request 2: the total number of candidates in the system is 500; the total number of candidates in the system with this specialization is 23; generation time – 0.3 seconds.

The results of generating recommendations for Request 3: the total number of candidates in the system is 500; the total number of candidates in the system with this specialization is 31; generation time – 0.4 seconds.

The average recommendation generation time is 0.4 seconds on 50 experiments. We believe that this is a very acceptable result, since during such a period it is possible to generate 10 candidates who potentially fit the given parameters of the request and are selected in comparison with candidates who have already been hired for similar requests. Which significantly speeds up the process of searching for candidates compared to manual processing of a large volume of data. The average time of a cursory review of one resume by a recruiter takes from about 12 seconds to several minutes [14].

Conclusions. The developed software can generate information for the selection of medical professionals based on already existing historical data of past requests from other customers. The developed site can be used for real recruiting systems.

In the future, the improvement of the implemented software can go along the path of adding the following functions:

- allow candidates to register themselves in the system and fill in their data and apply for vacancies;
- add the ability to integrate with third-party systems for data import and export;
- add the ability to recognize photos and documents to automatically extract data and fill in candidate profiles.

References

1. Jones M. T. *Recommender systems, Part 1. Introduction to approaches and algorithms.* URL: https://api.semanticscholar.org/CorpusID:13864749 (access date: 15.11.2023).
2. Codina V. A. *Recommendation System for the Semantic WEB*. URL: http://www.lsi.upc.edu/~vcodina/papers/dcai10 (access date: 07.10.2023).
3. Chandrashekhar H., Bhasker B. *Personalized recommender system using entropy-based collaborative filtering technique*. URL: https://www.proquest.com/docview/889977175 (access date: 05.10.2023).
4. Smith B., Linden G. *Two Decades of Recommender Systems at Amazon.com*. URL: https://ieeexplore.ieee.org/document/7927889 (access date: 20.09.2023).
5. *Singular Value Decomposition (SVD) & Its Application in Recommender System*. URL: https://analyticsindiamag.com/singular-value-decomposition-svd-application-recommender-system/ (access date: 10.10.2023).
6. Seeda P. A. *Complete Guide to Recommender Systems – Tutorial with Sklearn, Surprise, Keras, Recommenders*. URL: https://towardsdatascience.com/a-complete-guide-to-recommender-system-tutorial-with-sklearn-surprise-keras-recommender-5e52e8ceace1 (access date: 25.09.2023).
7. Aditya S., Rajora M., Singh I. *Hybrid Recommendation System for Item Cold Start Problem*. URL: https://doi.org/10.1109/SMARTGENCON51891.2021.9645839 (access date: 7.10.2023).
8. Narducci, F., Lops, P., Semeraro, G. *Power to the patients: The HealthNet social network*. URL: https://doi.org/10.1016/j.is.2017.07.005 (access date: 12.10.2023).
9. Han Q., Martinez de Rituerto de Troya I., Ji M., Gaur M., Zejnilovic L. *A Collaborative Filtering Recommender System in Primary Care: Towards a Trusting Patient-Doctor Relationship*. URL: https://doi.org/10.1109/ICHI.2018.00062 (access date: 13.10.2023).
10. Waqara M., Majeed M., Dawood H, Daud A. *An adaptive doctor-recommender system*. URL: http://dx.doi.org/10.1080/0144929X.2019.1625441 (access date: 14.10.2023).
11. *Implementing the Unit of Work Pattern in Clean Architecture with .NET Core*. URL: https://medium.com/@edin.sahbaz/implementing-the-unit-of-work-pattern-in-clean-architecture-with-net-core-53efb7f9d4d (access date: 03.11.2023).
12. *Everything you need to know about MVC architecture*. URL: https://towardsdatascience.com/everything-you-need-to-know-about-mvc-architecture-3c827930b4c1 (access date: 29.10.2023).
13. *Tutorial: Create a web API with ASP.NET Core*. URL: https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api (access date: 04.11.2023).
14. *Do Recruiters Really Only Spend 6 Seconds Reading Your Resume?* URL: https://newsletter.jobsearch.guide/p/do-recruiters-really-spend-6-seconds (access date: 14.11.2023).

References (transliterated)

1. Jones M. T. *Recommender systems, Part 1. Introduction to approaches and algorithms.* Available at:

https://api.semanticscholar.org/CorpusID:13864749 (accessed 15.11.2023).
2. Codina V. A. *Recommendation System for the Semantic WEB*. Available at: http://www.lsi.upc.edu/~vcodina/papers/dcai10 (accessed 07.10.2023).
3. Chandrashekhar H., Bhasker B. *Personalized recommender system using entropy-based collaborative filtering technique*. Available at: https://www.proquest.com/docview/889977175 (accessed 05.10.2023).
4. Smith B., Linden G. *Two Decades of Recommender Systems at Amazon.com*. Available at: https://ieeexplore.ieee.org/document/7927889 (accessed 20.09.2023).
5. *Singular Value Decomposition (SVD) & Its Application in Recommender System*. Available at: https://analyticsindiamag.com/singular-value-decomposition-svd-application-recommender-system/ (accessed 10.10.2023).
6. Seeda P. A. *Complete Guide to Recommender Systems – Tutorial with Sklearn, Surprise, Keras, Recommenders*. Available at: https://towardsdatascience.com/a-complete-guide-to-recommender-system-tutorial-with-sklearn-surprise-keras-recommender-5e52e8ceace1 (accessed 25.09.2023).
7. Aditya S., Rajora M., Singh I. *Hybrid Recommendation System for Item Cold Start Problem*. Available at: https://doi.org/10.1109/SMARTGENCON51891.2021.9645839 (accessed 07.10.2023).
8. Narducci, F., Lops, P., Semeraro, G. *Power to the patients: The HealthNet social network*. Available at: https://doi.org/10.1016/j.is.2017.07.005 (accessed 12.10.2023).
9. Han Q., Martinez de Rituerto de Troya I., Ji M., Gaur M., Zejnilovic L. *A Collaborative Filtering Recommender System in Primary Care: Towards a Trusting Patient-Doctor Relationship*. Available at: https://doi.org/10.1109/ICHI.2018.00062 (accessed 13.10.2023).
10. Waqara M., Majeed M., Dawood H, Daud A. *An adaptive doctor-recommender system*. Available at: http://dx.doi.org/10.1080/0144929X.2019.1625441 (accessed 14.10.2023).
11. *Implementing the Unit of Work Pattern in Clean Architecture with .NET Core*. Available at: https://medium.com/@edin.sahbaz/implementing-the-unit-of-work-pattern-in-clean-architecture-with-net-core-53efb7f9d4d (accessed 03.11.2023).
12. *Everything you need to know about MVC architecture*. Available at: https://towardsdatascience.com/everything-you-need-to-know-about-mvc-architecture-3c827930b4c1 (accessed 29.10.2023).
13. *Tutorial: Create a web API with ASP.NET Core*. Available at: https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api (accessed 04.11.2023).
14. *Do Recruiters Really Only Spend 6 Seconds Reading Your Resume?* Available at: https://newsletter.jobsearch.guide/p/do-recruiters-really-spend-6-seconds (accessed 14.11.2023).

Received 20.11.2023

УДК 004.9

В. В. ЯМБУРЕНКО, Національний технічний університет «Харківський політехнічний інститут», студент, м. Харків, Україна, e-mail: yamburenko@gmail.com, ORCID: https://orcid.org/0009-0005-4797-8684
І. В. ЛЮТЕНКО, кандидат технічних наук (PhD), доцент, Національний технічний університет «Харківський політехнічний інститут», доцент кафедри програмної інженерії та інтелектуальних технологій управління; м. Харків, Україна, e-mail: cherliv68@gmail.com, ORCID: https://orcid.org/0000-0003-4357-1825
А. М. КОПП, доктор філософії (PhD), доцент, Національний технічний університет «Харківський політехнічний інститут», доцент кафедри програмної інженерії та інтелектуальних технологій управління; м. Харків, Україна, e-mail: kopp93@gmail.com, ORCID: https://orcid.org/0000-0002-3189-5623
Д. М. ГОЛОВНЯ, Національний технічний університет «Харківський політехнічний інститут», студент, м. Харків, Україна, e-mail: dimagolovnya@gmail.com, ORCID: https://orcid.org/0009-0009-0901-025X

АЛГОРИТМ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПІДБОРУ МЕДИЧНОГО ПЕРСОНАЛУ

В роботі будь-якої організації є багато рутинної роботи, в тому числі і в агентствах по найму персоналу. Ефективна організація управління та автоматизації діяльності співробітників рекрутингових агентств – непросте завдання. Система повинна автоматизувати рутинні дії робітників рекрутингових агентств та бути зручною для їх клієнтів. Ця робота пропонує підхід для автоматизації роботи по підбору необхідного медичного персоналу. Не всі інформаційні системи, що використовуються рекрутинговими агентствами можуть робити порівняння кандидатів та генерувати пропозиції, які включають декілька найкращих кандидатів. На підставі аналізу предметної області були визначені групи параметрів, які суттєво впливають на вибір медичного персоналу. Запропонований підхід полягає у тому, щоб проаналізувати запит від клієнта, а потім віднайти в системі схожі на нього за параметрами запити інших клієнтів, по яким вже було знайдено кандидата. Наступним кроком береться профілі медичних працівників, що були запропоновані для цих запитів (вони виступають як еталонні) для подальшого порівняння їх із існуючими кандидатами. Для кожного параметра профілю працівника є своя функція схожості. Наявні кандидати отримають оцінки, після чого відбувається їх ранжування. Також додатково проводимо корегування оцінки шляхом порівняння кандидатів із поточним запитом. Для автоматизації діяльності по підбору медичного персоналу було розроблене програмне забезпечення. Для його реалізації запропонована трьохрівнева клієнт серверна архітектура. Для серверної частини було обрано MVC (Model View Controller) архітектуру. Для клієнтської частини використаний архітектурний шаблон Single Page Application. Серверна частина розбита на три шари, які додатково розмежовують та структурують відповідальність компонентів системи. Для реалізації бізнес логіки використано технології .NET. Для сервера та провайдеру баз даних використано SQL Server. Використання програмної реалізації розробленої системи продемонструвало досить непогані результати. Середній час підбору 10 кращих кандидатів з 500 становить 0,4 секунди, а обробка тільки 1 резюме людиною займає декілька хвилин.

Ключові слова: підбор персоналу, автоматизація, обробка даних, алгоритм, програмне забезпечення, системна архітектура, технології розробки.

Повні імена авторів / Author's full names

Автор 1 / Author 1: Ямбуренко Віктор Вікторович, Yamburenko Viktor Viktorovych
Автор 2 / Author 2: Лютенко Ірина Вікторівна, Liutenko Iryna Viktorivna
Автор 3 / Author 3: Копп Андрій Михайлович, Kopp Andrii Mykhailovych
Автор 4 / Author 4: Головня Дмитро Миколайович, Holovnia Dmytro Mykolaiovych