

Intelligent System for Socialization by Personal Interests on the Basis of SEO Technologies and Methods of Machine Learning

Taras Batiuk^[0000-0001-5797-594X], Victoria Vysotska^[0000-0001-6417-3689], Vasyly Lytvyn^[0000-0002-9676-0180]

Lviv Polytechnic National University, Lviv, Ukraine

tbatjuk4u@gmail.com

Abstract. Creating of an information system that would allow efficient analysis and selection of users is an important task as there are currently no such systems. Socialization system will optimize the process of finding and dating people on the Internet. To implement this system, we need to use a convolutional neural network to find a human face in the photo. Fuzzy search algorithms and the Noisy Channel model are also required to analyze user information and list relevant users.

Keywords: Levenshtein distance, Convolutional neural network, Fuzzy search, Noisy Channel.

1 Introduction

The main purpose of this study is to create an information system for the socialization of individuals using fuzzy word search algorithms using the Noisy Channel model with algorithms for efficient textual information distribution, and using a convolutional neural network to identify system users, since there is no such system at present, that would analyze the user-specified information and form a list of the most relevant users.

Creating an information system for the socialization of personalities is an urgent task, since in the modern world, people try to optimize all life processes to save time and, accordingly, for a more expedient realization of time saved. When looking for programs, users first of all choose the ones that save time, are optimized and automated enough to do most of the work instead of the system user. This information system combines at once two important tasks: socialization of users and optimizes and automates the process of socialization as much as possible.

There are quite a number of articles on this topic, for example, in [1] and [2] the authors propose a new ranking system among social network users and an adaptive network model for easy interaction between users, and the authors propose to improve the system of aggregation and sorting of information on social networks, according to the set parameters.

Copyright © 2020 for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2 Analysis of Literary Sources and Subject Field

In articles [3] and [4], the authors propose to improve social networking capabilities in terms of analyzing user data and developing user performance, and to improve social networking capabilities in terms of finding commonalities between users and, subsequently, identifying the main interests of the user and giving the user the most relevant matches in your subsequent search for information on the social network.

Also in article [5], the authors propose that the authors propose a new approach for sorting and storing data in the cloud and issuing it when searching using a new flexible data delivery model on social networks [6-13].

First of all, when it comes to the advantages of the information system over analogs, it is worth noting that there are almost no analogues with a similar system of work, among which there are Tinder and Badoo programs [14-19]. The similarity of my system to them in the use of a convolutional neural network, which checks for the presence of similar users and is used to identify users of the system [20-28].

The main advantage is that Tinder and Badoo provide the most limited socialization mechanism, allowing users to filter by gender, age and location, which is not optimized and does not save much time on socialization, so we need to spend too much time for talking with users and we are not sure, that user has the same interests with us and because of it we can just waste our time [29-42]. Our system analyzes the user information and on that basis creates a list of users sorted by the calculated percentage of similarity between the users of the system, which is the main advantage of the system compared to its main analogues.

3 Specifying the Functioning of the System

The information system should consist of the following steps: registration or authorization of the user, familiarization of the user with the rules of using the system. Uploading a photo that will be checked using a convolutional neural network for the presence of a human face and for the presence of that person in an existing database, adding a user avatar and filling the album with photos.

The next step is to provide the user with information about himself, to analyze the information and to compare it with the information about the already existing ones in the system using the Levenstein algorithm, fuzzy search algorithms and the Noisy Channel model, after which a list of users will be sorted by the calculated percentage of similarity between users.

It is possible to further filter the generated list, change the personal information resulting in the newly created list, or go to the main menu of the system. Also, any changes made by the user and correspondence will be stored in the database, and all photos will be stored in the cloud media storage service.

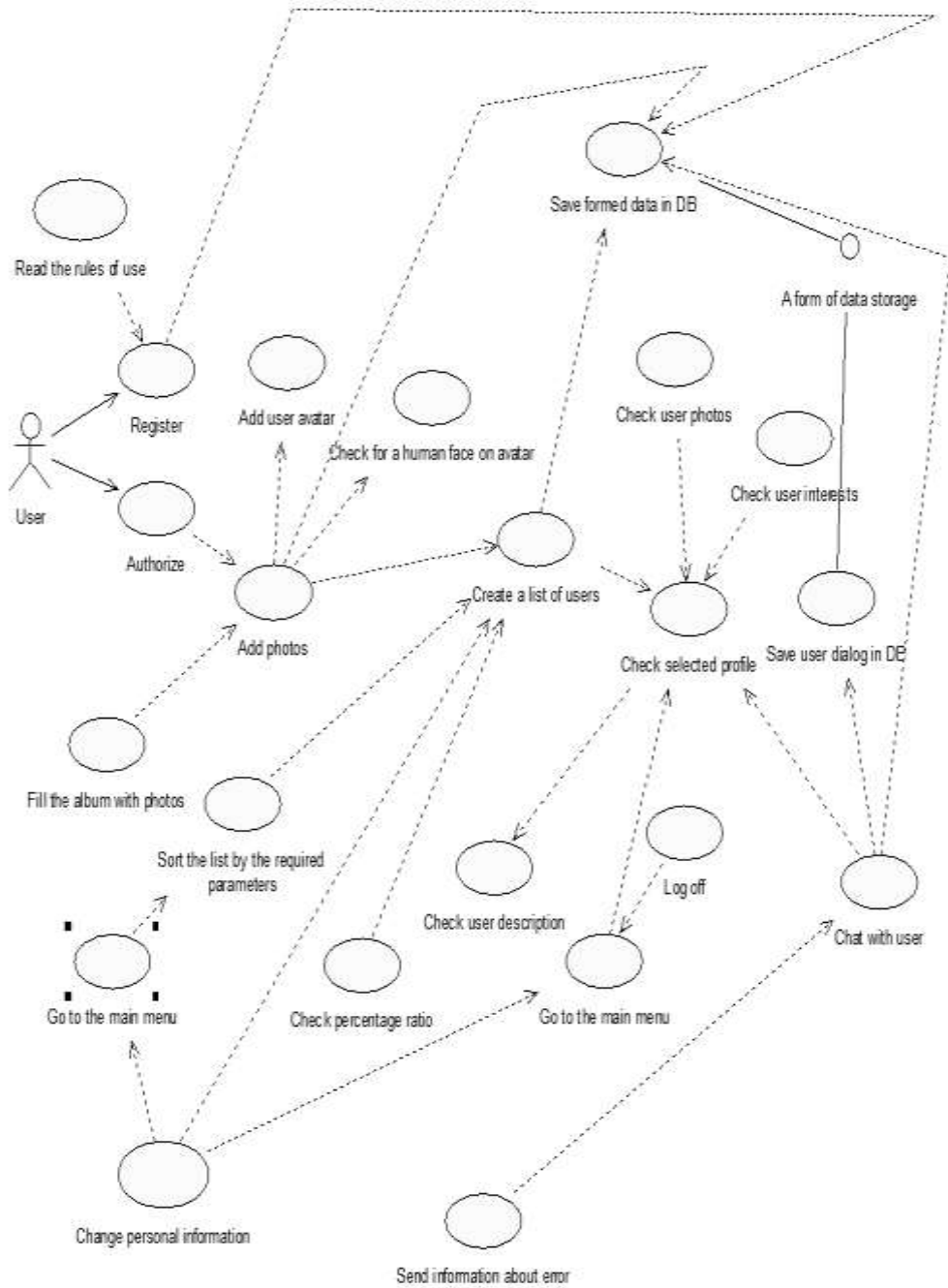


Fig.1 Use Case digram

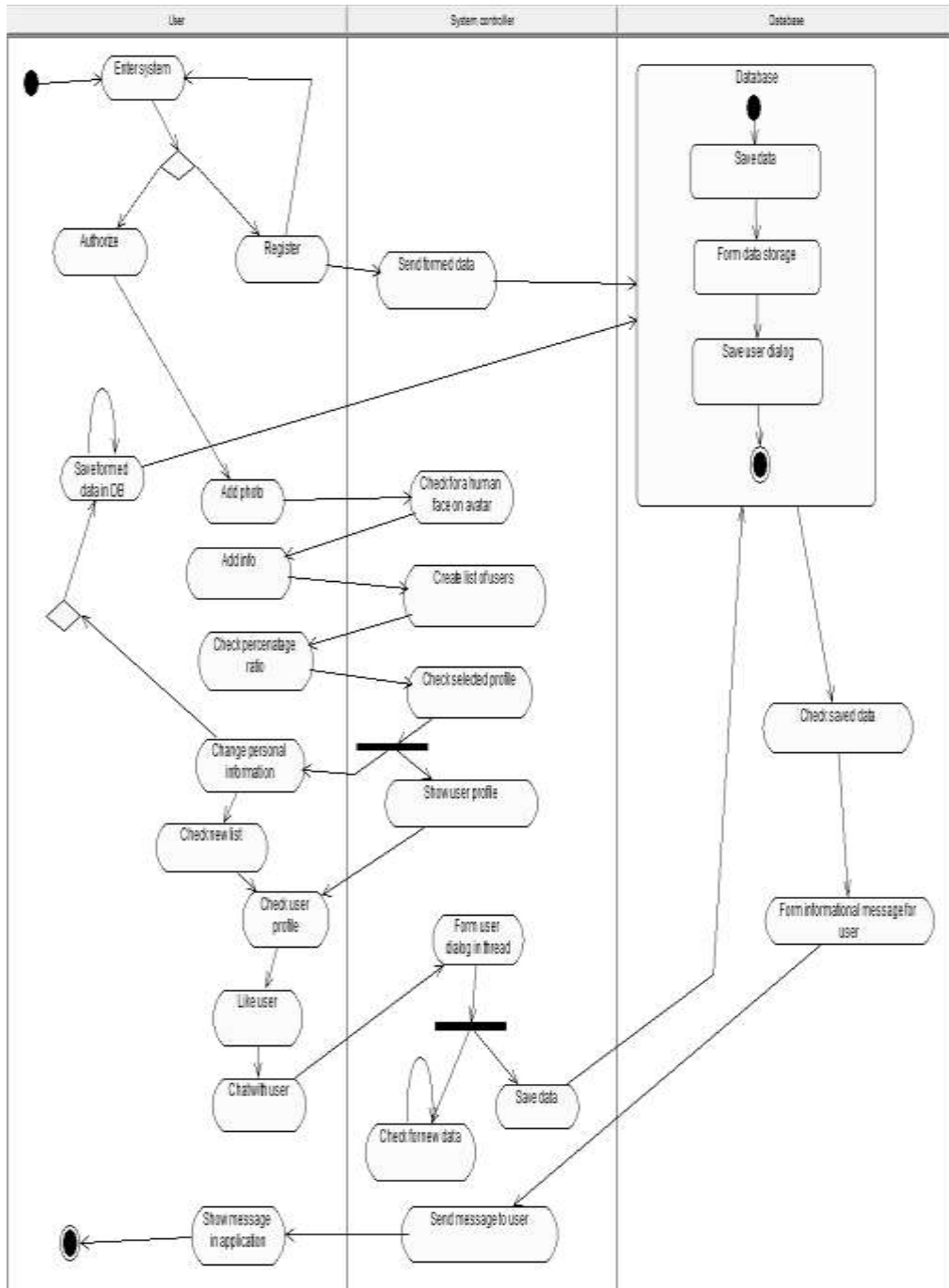


Fig.2 Activity diagram

Separate storage in the database can be made using ORM, which avoids direct work with the database. Which could lead to some inaccuracies at the base level, in this case I'm working with objects, which allows retrieve by accessing a list of objects.

The last step is to research the profile and interests of the selected from the selected user sample and start communicating with the selected user, all correspondence in the form of a separate message flow will be stored in the database. The general algorithm of operation of the information system is shown in figure 1 in the form of use case diagram and in figure 2 in the form of activity diagram.

4 Software to Solve the Problem

The following are basic screenshots of the information system, figure 3 shows the process of adding information about the user, namely the description, interests and general information about what and what the user is looking for.

Fig.3 User information

For adding data I use reactive forms. Reactive forms use an explicit and immutable approach to managing the state of a form at a given point in time. Each change to the form state returns a new state, which maintains the integrity of the model between

changes. Reactive forms are built around observable streams, where form inputs and values are provided as streams of input values, which can be accessed synchronously.

Reactive forms also provide a straightforward path to testing because you are assured that your data is consistent and predictable when requested. Any consumers of the streams have access to manipulate that data safely.

Reactive forms differ from template-driven forms in distinct ways. Reactive forms provide more predictability with synchronous access to the data model, immutability with observable operators, and change tracking through observable streams.

If you prefer direct access to modify data in your template, template-driven forms are less explicit because they rely on directives embedded in the template, along with mutable data to track changes asynchronously.

Figure 4 depicts the process of adding user photos manually or through a guide, 1 or more photos, as well as stopping and deleting unnecessary photos while adding them.

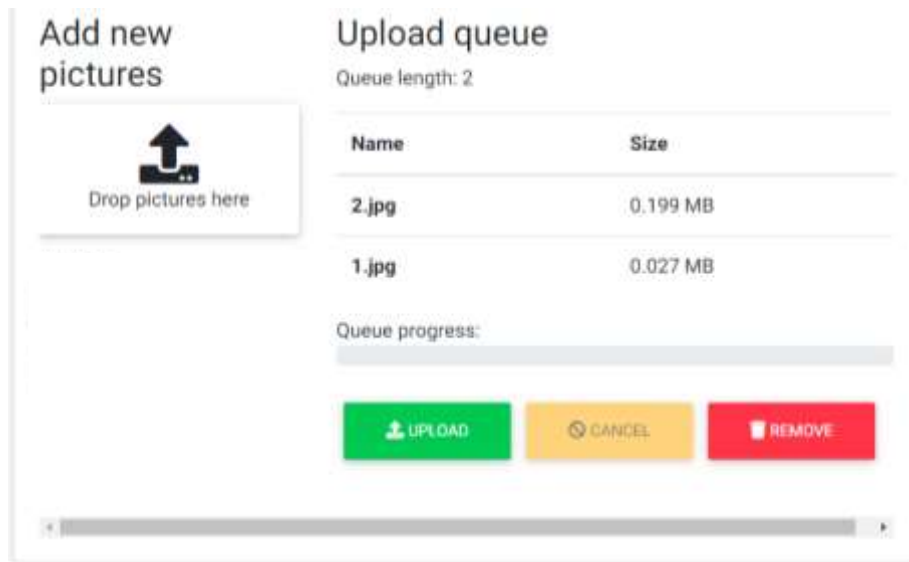


Fig.4 The process of adding photos

Figure 5 shows the currently added photos and the use of the convolutional neural network. A convolutional neural network consists of layers of two types: convolution layers and subsampling layers or sublayer layers. In it, each layer has a topographic structure, that is, each neuron is connected to a fixed point of the original image, as well as to the region of the input image that is processed by the given neuron.

Obviously, the first thing to do would be to pass in an image to the program. In this model, we want to create an image pyramid, in order to detect faces of all different sizes. In other words, we want to create different copies of the same image in different sizes to search for different sized faces within the image.

Having a stride of 2 helps reduce computation complexity without significantly sacrificing accuracy. Since faces in most images are significantly larger than two pixels, it's highly improbable that the kernel will miss a face merely because it shifted 2 pixels.

At the same time, your computer (or whatever machine is running this code) will have a quarter of the amount of operations to compute, making the program run faster and with less memory. There are a number of different neurons in each location of each layer, each with its own set of input weights associated with the neurons in the rectangular fragment of the previous layer. A convolutional neural network allows you to determine whether such a user is already in the database, and whether the face is depicted in the photo, as seen in the selected fragment of the screenshot - 2 pictures the system does not allow to put on the avatar, because they do not have a face on figure 5.

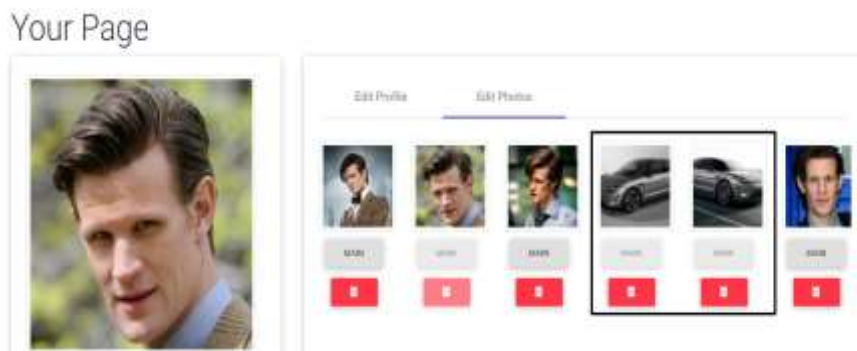


Fig.5 Choosing a user avatar

Figure 6 shows a generated list of users using fuzzy search sorted by calculated percentage of similarity between users. Fuzzy search algorithms have chosen linear search - a simple, consistent application of a metric (in this case, a Levenstein metric) to words in the input text.

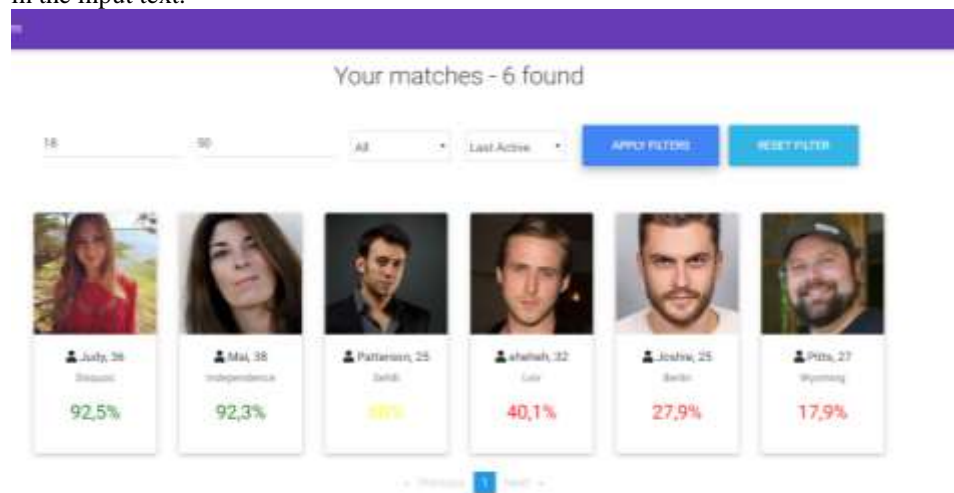


Fig.6 A list of users has been generated

To calculate the optimal distance between words in the text, you need to modify the algorithm for finding the ordinary Levenshtein distance as follows: store not the last two but three rows of the matrix, and add the appropriate additional condition - in the

case of detecting transposition in the calculation of the distance also take into account its cost. Thus, by modifying the algorithm, the Noisy Channel model was used, which made it possible to optimize the information analysis process, first by creating a user parameter matrix, then asynchronously using linear search to compare the current parameters and form a percentage of similarity for each parameter. The last step will be to find the average percentage of user similarities.

Figure 7 depicts a user's choice of which coincided by 92.5 percent, clicking on the image brings a photo of the user we choose and gives the opportunity to choose one of 3 options: view the user's profile to see his interests and find out all the necessary information about him, put Like to save the link to this user and not further lose it in the user list, or write a private message and start a conversation with the user. All buttons are dynamically loaded into the tree using a special directive that optimizes the system. Also, each user image is a separate component, which is loaded into the main window of the program by means of decorators.



Fig.7 User choice

A decorator is a special kind of declaration that can be attached to a class declaration, method, accessor, property, or parameter. Decorators use the form `@expression`, where expression must evaluate to a function that will be called at runtime with information about the decorated declaration.

For numbers, in the event you wish all your currency values to adhere to 2 decimal places, you could apply a `@rounded` decorator to these params as an alternative to poll your boilerplate with primitive Javascript functions.

This sums up the current state of decorator implementations. At this stage you may be forming an idea of where you design like decorators applied - is it worth it to introduce additional complexity to your method signatures with parameter decorators? Or you would lean towards just using decorators to modify methods only - this will boil down to either a personal taste or a collective team decision from project to project.

Decorators can be used to inject class properties, aid in collaborative development by annotating issues at certain points in your codebase - whether at the class level or

method level - or simply log method arguments and return values in a non-obstructive way. Decorators follow the decorator design pattern, and are often referred to as a type of meta programming. In the realm of Javascript, Angular have heavily adopted decorators into the framework. We have also seen packages such as lodash adopt decorators to easily apply its functions to your code using the feature.

Figures 8 and 9 show the likes of us and the likes of us. This is very handy and even if we have not written anything in private messages, we still have a link to the system user and will not lose it among a large number of other users.

Also, the user to whom we put a Like learns about it and can write to us in private messages. Also, if someone has liked us, we will be able to see it, see the profile of the user, his main interests, view the photos and write to the user who has chosen us, so that the opportunity to communicate effectively.

Each user is a separate component of the system, so we use the functionality of the same components, optimizing the system and not reloading the page.

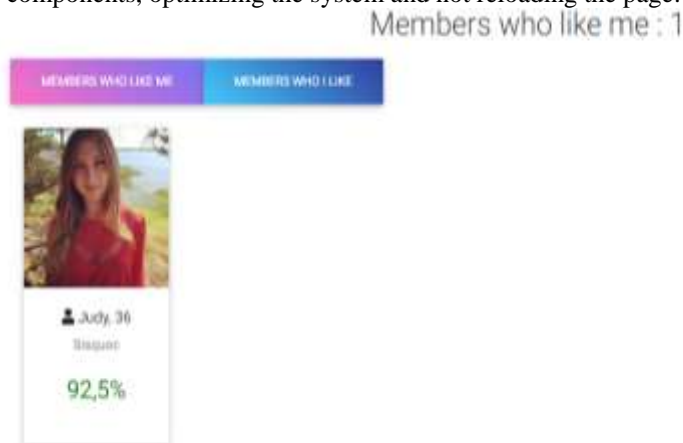


Fig.8 The users who have chosen us

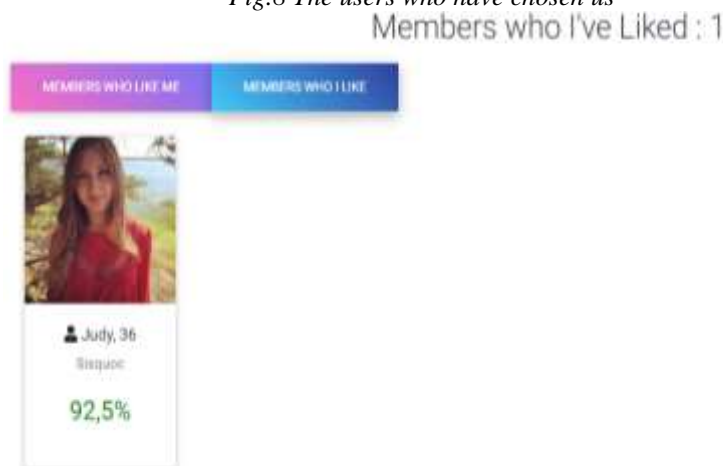


Fig.9 The users we chose

Figures 10 and 11 show basic information about the selected user as tabs for maximum usability. Tabs organize content into separate views where only one view can be visible at a time. Each tab's label is shown in the tab header and the active tab's label is labeled with the animated ink bar. When the list of tab labels exceeds the width of the header, pagination controls appear to let the user scroll left and right across the labels. By default, the tab contents are eagerly loaded. Eagerly loaded tabs will activate the child components but will not inject them into the DOM until the tab is activated. If the tab contains several complex child components or the contents of the tab rely on DOM calculations during initialization, it is advised to lazy load the tab's content.

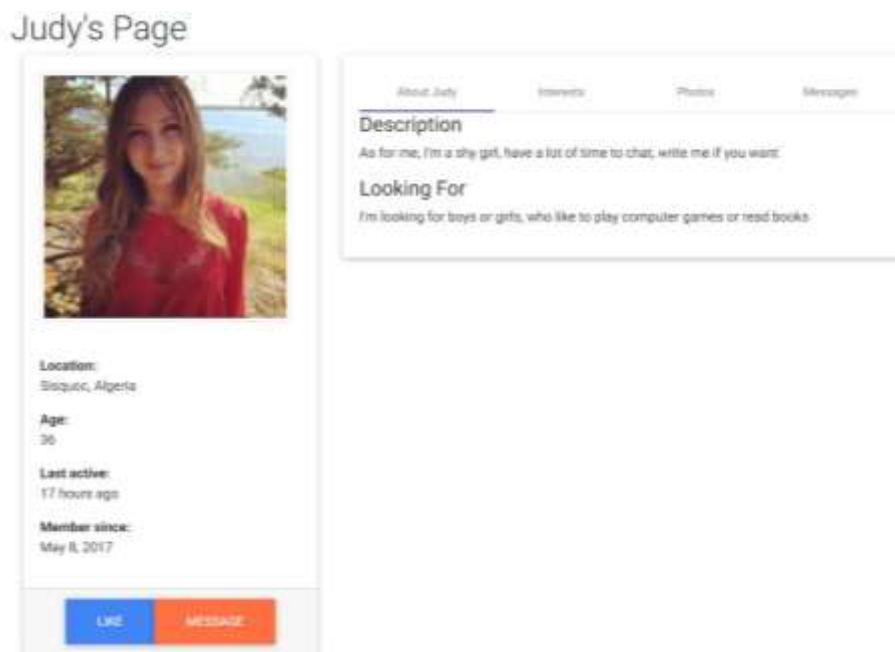


Fig.10 About user



Fig.11 User interests

Figures 12 and 13 show an album view of the selected user and private correspondence with the user. As can be seen from Figure 13, the user's avatar, the time of sending a private message and information about the read or unread message are displayed in the correspondence with the user, which allows effective and convenient correspondence with different users of the system.

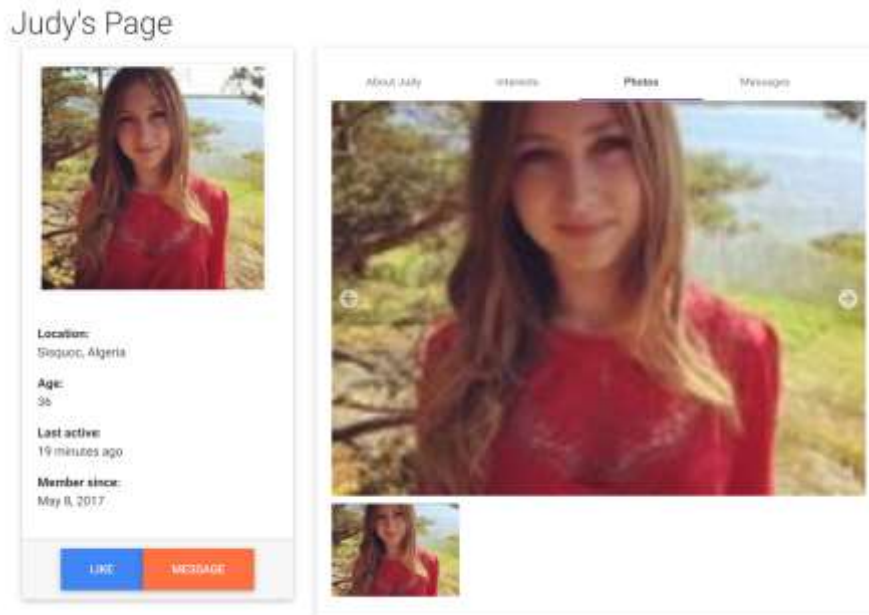


Fig.12 User's photos view

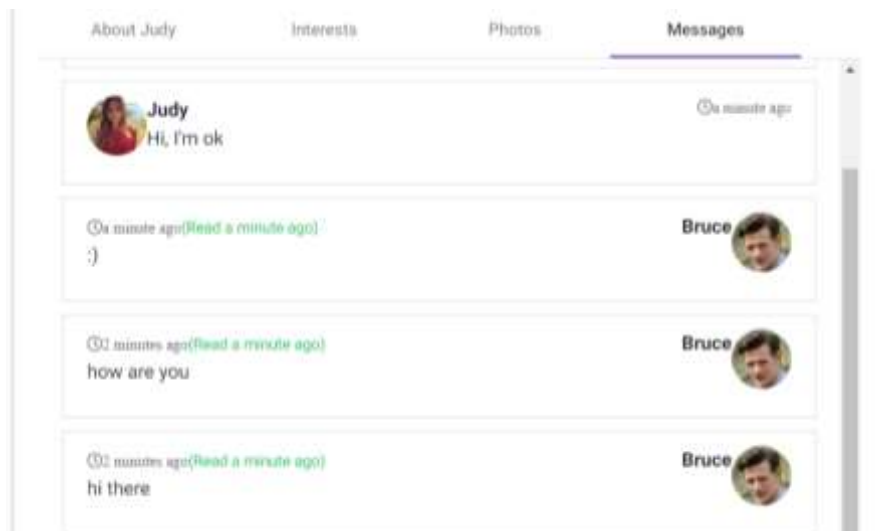


Fig.13 Chatting with selected user

5 Conclusions

In today's world, users are first and foremost looking for a user-friendly and intuitive system that can perform certain actions for them, thereby saving users time. The system for socialization of personal interests of the common interests overtakes the programs of analogues, because in the first place it guarantees the identification and authenticity of a photo of a user and also it automates and optimizes the process of finding users by analyzing all users on a common interest and forming a percentage ratio that allows you to quickly determine your degree of similarity with other users and have an approximate idea of how much your interests are in line with those of others.

References

1. Parry, M. E., Kawakami, T., Kishiya, K.: The effect of personal and virtual word-of-mouth on technology acceptance. In: *Journal of Product Innovation Management*, 29(6), 952–966 (2012). <https://onlinelibrary.wiley.com/toc/15405885/2012/29/6>.
2. Schivinski, B., & Dąbrowski, D.: The effect of social-media communication on consumer perceptions of brands. In: *Journal of Marketing Communications*, 22(2), 189–214. (2014).
3. Ranjbaran, B., Jamshidian, M., & Dehghan, Z.: A survey of identification of major factors influencing customers attitude toward machine made carpet brands. In: *Journal of Business Strategies*, 5(23), 109–118. (2007).
4. Schmäh, M., Wilke, T., & Rossmann, A.: Electronic word of mouth: In: A systematic literature analysis. *Digital Enterprise Computing*, 147–158. (2017).
5. Wang, X., Yu, C., & Wei, Y.: Social media peer communication and impacts on purchase intentions: A consumer socialization framework. In: *Journal of Interactive Marketing*, 26(4), 198–208. (2012).
6. Lytvyn, V., Vysotska, V., Rzhеuskyi, A.: Technology for the Psychological Portraits Formation of Social Networks Users for the IT Specialists Recruitment Based on Big Five, NLP and Big Data Analysis. In: *CEUR Workshop Proceedings*, Vol-2392, 147–171. (2019)
7. Lytvyn, V., Vysotska, V., Mykhailyshyn, V., Rzhеuskyi, A., Semianchuk, S.: System Development for Video Stream Data Analyzing. In: *Advances in Intelligent Systems and Computing*, 1020, 315–331. (2020)
8. Lytvyn, V., Vysotska, V., Peleshchak, I., Peleshchak, R., Kohut, I.: Intelligent system of a smart house. In: *3rd International Conference on Advanced Information and Communications Technologies*, AICT, 282–287. (2019)
9. Kravets, P., Burov, Y., Lytvyn, V., Vysotska, V.: Gaming method of ontology clusterization. In: *Webology*, 16(1), 55–76. (2019)
10. Lytvyn, V., Gozhyj, A., Kalinina, I., Vysotska V., Shatskykh, V., Chyrun, L., Borzov, Y.: An intelligent system of the content relevance at the example of films according to user needs. In: *CEUR Workshop Proceedings*, Vol-2516, 1–23. (2019)
11. Gozhyj, A., Kalinina, I., Gozhyj, V., Vysotska, V.: Web service interaction modeling with colored petri nets. In: *International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, IDAACS, 1, 319–323. (2019)

12. Babichev, S.: An Evaluation of the Information Technology of Gene Expression Profiles Processing Stability for Different Levels of Noise Components. In: *Data*, 3 (4), art. no. 48. (2018)
13. Babichev, S., Durnyak, B., Pikh, I., Senkivskyy, V.: An Evaluation of the Objective Clustering Inductive Technology Effectiveness Implemented Using Density-Based and Agglomerative Hierarchical Clustering Algorithms. In: *Advances in Intelligent Systems and Computing*, 1020, 532-553. (2020)
14. Boreiko, O. Y., Teslyuk, V. M., Zelinskyy, A., Berezhsky, O.: Development of models and means of the server part of the system for passenger traffic registration of public transport in the "smart" city. In: *Eastern-European Journal of Enterprise Technologies*, 1(2-85), 40-47. (2017)
15. Kazarian, A., Teslyuk, V., Tsmots, I., Tykhan, M.: Implementation of the Face Recognition Module for the "Smart" Home Using Remote Server. In: *Advances in Intelligent Systems and Computing III, AISC 871*, Springer, 17–27. (2019)
16. Rusyn, I. B., Valko, B. T.: Container landscaping with *Festuca arundinaceae* as a mini bioelectrical systems in a modern buildings. In: *International Journal of Energy for a Clean Environment*, 20(3), 211-219. (2019).
17. Rusyn, I. B., Hamkalo, Kh. R.: Bioelectricity production in an indoor plant-microbial biotechnological system with *Alisma plantago-aquatica*. In: *Acta Biologica Szegediensis*, 62 (2), 170-179. (2018).
18. Anisimova, O., Vasylenko, V., Fedushko, S.: Social Networks as a Tool for a Higher Education Institution Image Creation. In: *CEUR Workshop Proceedings, Vol-2392*, 54–65. (2019)
19. Kanishcheva, O., Cherednichenko, O., Sharonova, N.: Image Tag Core Generation. In: *CEUR Workshop Proceedings, Vol-2533*, 35-44. (2019)
20. Korzh, R., Fedushko, S., Peleschychyn, A.: Methods for forming an informational image of a higher education institution. In: *Webology*, 12(2), Article 140. (2015)
21. Korzh, R., Peleschychyn, A., Syerov, Yu., Fedushko, S.: Principles of University's Information Image Protection from Aggression. In: *Proceedings of the XIth International Scientific and Technical Conference (CSIT)*, 77-79. (2016)
22. Korzh, R., Peleschychyn, A., Fedushko, S., Syerov, Y.: Protection of University Information Image from Focused Aggressive Actions. In: *Advances in Intelligent Systems and Computing*, Springer International Publishing, 543, 104-110. (2017)
23. Korzh, R., Peleschychyn, A., Syerov, Yu., Fedushko, S.: University's Information Image as a Result of University Web Communities' Activities. In: *Advances in Intelligent Systems and Computing*, Springer International Publishing, 512, 115-127. (2017)
24. Veres, O., Rusyn, B., Sachenko, A., Rishnyak, I.: Choosing the method of finding similar images in the reverse search system. In: *CEUR Workshop Proceedings*, 2136, 99-107. (2018)
25. Varetskyy, Y., Rusyn, B., Molga, A., Ignatovych, A.: A new method of fingerprint key protection of grid credential. In: *Advances in Intelligent and Soft Computing*, 84, 99-103. (2010)
26. Basyuk, T., Vasyliuk, A., Lytvyn, V.: Mathematical Model of Semantic Search and Search Optimization. In: *CEUR Workshop Proceedings, Vol-2362*, 96-105. (2019)
27. Rusyn, B., Kosarevych, R., Lutsyk, O., Korniy, V.: Segmentation of atmospheric clouds images obtained by remote sensing. In: *14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET*, 213-216. (2018)

28. Kapustiy, B.O., Rusyn, B.P., Tayanov, V.A.: A new approach to determination of correct recognition probability of set objects. In: *Upravlyaushchie Sistemy i Mashyny*, 2, 8-12. (2005)
29. Rusyn, B., Prudyus, I., Ostap, V.: Fingerprint image enhancement algorithm. In: *The Experience of Designing and Application of CAD Systems in Microelectronics, Proceedings of the 6th International Conference ,CADSM*, 193-194. (2001)
30. Rusyn, B., Torska, R., Kobasyar, M.: Application of the cellular automata for obtaining pitting images during simulation process of thei growth. In: *Advances in Intelligent Systems and Computing*, 242, 299-306. (2014)
31. Oksanych, I., Shevchenko, I., Shcherbak, I., Shcherbak, S.: Development of specialized services for predicting the business activity indicators based on micro-service architecture. In: *Eastern-European Journal of Enterprise Technologies*, 2(2-86), 50-55. (2017)
32. Galushka, I., Shcherbak, S.: Devising a mathematical model for pattern-based enterprise data integration. In: *Eastern-European Journal of Enterprise Technologies*, 2(9), 59-64. (2015)
33. Shcherbak, S.S.: Interoperability web application models based on microformats. In: *CriMiCo, 21st International Crimean Conference: Microwave and Telecommunication Technology*, 57-58. (2011)
34. Nazarkevych, M., Klyujnyk, I., Maslanych, I., Havrysh, B., Nazarkevych, H.: Image filtration using the Ateb-Gabor filter in the biometric security systems. In: *International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH)*, 276-279. (2018)
35. Nazarkevych, M., Buriachok, V., Lotoshynska, N., Dmytryk, S.: Research of Ateb-Gabor Filter in Biometric Protection Systems. In: *International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, 1, 310-313. (2018)
36. Nazarkevych, M., Oliarnyk, R., Dmytruk, S.: An images filtration using the Ateb-Gabor method. In: *International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, 1, 208-211. (2017)
37. Lytvyn, V., Peleshchak, I., Peleshchak, R.: The compression of the input images in neural network that using method diagonalization the matrices of synaptic weight connections. In: *2nd International Conference on Advanced Information and Communication Technologies, AICT*, 66-70. (2017)
38. Lytvyn, V., Peleshchak, I., Peleshchak, R., Holoshchuk, R.: Detection of multispectral input images using nonlinear artificial neural networks. In: *14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET*, 119-122. (2018)
39. Shakhovska, N., Nych, L., Kaminskyj, R.: The identification of the operator's systems images using the method of the phase portrait. In: *Advances in Intelligent Systems and Computing* 512, 241-253. (2017)
40. Kravets, P.: The control agent with fuzzy logic. In: *Perspective Technologies and Methods in MEMS Design, MEMSTECH*, 40-41. (2010)
41. Kravets, P.: The game method for orthonormal systems construction. In: *The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM*, 296-298. (2007)
42. Kravets, P., Kyrkalo, R.: Fuzzy logic controller for embedded systems. In: *International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH*, 58-59. (2009)