

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова
праця на правах рукопису

ПАПА ОЛЕКСАНДР АНДРІЙОВИЧ

УДК: 004.9:35

ДИСЕРТАЦІЯ

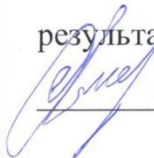
Моделювання ефективності веб-сервісів для надання
адміністративних послуг

121 – Інженерія програмного забезпечення

12 – Інформаційні технології

Подається на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело

 Папа О.А.

Науковий керівник Пукас Андрій Васильович, кандидат технічних наук,
доцент.

Тернопіль - 2021



АНОТАЦІЯ

Папа О.А. Моделювання ефективності веб-сервісів для надання адміністративних послуг - Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 121 «Інженерія програмного забезпечення» - Західноукраїнський національний університет, Тернопіль, 2020.

Підготовка здійснювалась на кафедрі комп'ютерних наук Західноукраїнського національного університету Міністерства освіти і науки України.

Дисертаційна робота присвячена розв'язуванню актуального науково-технічного завдання розробки математичного та програмного забезпечення підтримки веб-сервісів надання адміністративних послуг шляхом впровадження засобів інтелектуалізації процесу виконання послуг та математичного моделювання ефективності функціонування веб-сервісів.

У першому розділі проведено аналіз існуючих методів та засобів організації веб-сервісів та адміністративних послуг.

Проведений аналіз показав, що одним із визначальних чинників процесу створення ефективних ЦНАП та підвищення якості надання адміністративних послуг є використання сучасних інформаційних технологій. Ефективна інформаційна система надання адміністративних послуг дасть змогу одержувати об'єктивну інформацію щодо якості надання послуг. До того ж, така система може стати основою для оперативного ухвалення управлінських рішень з метою реального поліпшення та наближення рівня якості послуг до потреб замовників. Для цього переважно створюються мобільні додатки, які наближають до сервісів велику кількість користувачів. Разом з тим такі системи не зменшують часу надання послуг, оскільки вимагають достатньої обізнаності користувача щодо особливостей

отримання тих чи інших послуг, а отже потребують інтелектуалізації процесу виконання послуги.

Таким чином, актуальною є задача використання інтелектуальних методів обробки даних та математичних методів оцінювання ефективності веб-сервісів, щоб підвищити якість, а також значно прискорити надання адміністративних послуг для громадян.

Далі у цьому розділі проаналізовано відомі вітчизняні та зарубіжні програмні засоби для надання адміністративних послуг. Серед найвідоміших програмних засобів відмічено такі: iGov – портал державних послуг; дія - державні послуги онлайн; “Сервіс Канада”; “Офіси для громадян” - Франція; “WOM” - Польща.

Усі із перерахованих не містять інтелектуальних модулів або ці модулі мають обмежений функціонал.

Важливим питанням також є оцінювання ефективності функціонування веб-сервісів. Проведений аналіз літературних джерел показав, що переважно показники ефективності не стосуються безпосереднього функціонування веб-ресурсів, що є суттєвою проблемою в оцінюванні якості веб-сервісу.

Окремою проблемою є встановлення формалізованого опису показників ефективності веб-сервісів, який би давав можливість моделювання цієї ефективності для різних умов. Таким чином проблема моделювання ефективності функціонування веб-сервісу, яка є складовою для розв’язування вище зазначеної задачі, залишається актуальною в силу різноманіття цих сервісів та різноманіття процедур надання адміністративних послуг.

Виходячи із вищезазначеного, актуальною є науково-прикладна задача розробки математичного та програмного забезпечення підтримки веб-сервісів надання адміністративних послуг за рахунок впровадження засобів

інтелектуалізації процесу виконання послуг та математичного моделювання ефективності функціонування веб-сервісів.

Далі у розділі проведено обґрунтування засобів організації онлайн сервісів. Наведено метрики ефективності функціонування онлайн сервісів.

У завершальній частині розділу здійснено постановку завдань дисертаційного дослідження.

У другому розділі дисертаційної роботи вперше запропоновано та обґрунтовано показник ефективності функціонування веб-сервісів у вигляді динаміки відвідуваності користувачами та його математичну модель, яка на відміну від існуючих, побудована на методах аналізу інтервальних даних та урахування циклічності процесів відвідуваності, що забезпечило гарантований щоденний прогноз кількості запитів веб-сервісу. Розроблена інтервальна модель та метод її ідентифікації забезпечила, можливість планування та розподілу ресурсів організації щодо надання он-лайн послуг. Для побудови зазначеної моделі використано поєднання методів аналізу інтервальних даних та параметричної ідентифікації моделі. Зазначений метод, своєю чергою, побудовано на обчислювальних схемах з самоорганізацією за аналогією із поведінковими моделями бджолоїної колонії. Також у цьому розділі наведено методику отримання експериментальних даних із діючого веб-сервісу та розглянуто приклад ідентифікації інтервальної моделі ефективності цього веб-сервісу.

У третьому розділі розглянуто математичне та програмне забезпечення для побудови інтелектуального модуля у веб-сервісах надання адміністративних послуг. Запропоновано новий метод інтелектуального опрацювання інформації для оцінювання подібності он-лайн послуг, який на відміну від існуючих, ґрунтується на кластеризації документів, що оперують поняттями відстані між об'єктами- реквізитами. Запропонований метод

забезпечує зниження часу надання он-лайн послуг і в цілому забезпечує підвищення ефективності веб-сервісів.

Удосконалено метод автоматизованого пошуку оптимальних наборів документів в заданій предметній області, представлених напрямленим графом, який на відміну від існуючих використовує алгоритм мурашиної колонії, що забезпечило створення інтелектуального програмного модуля, який, своєю чергою, автоматизує процес надання адміністративної послуги і відповідно, скорочує час надання для окремих категорій послуг.

В завершальній частині цього розділу наведено алгоритмічне та програмне забезпечення реалізації інтелектуального модуля в складі веб-сервісів. Розроблено схему видалення дублюючих документів та послуг в процесі формування пошукового запиту на адміністративні послуги. Наведено результати розробки програмного інтерфейсу для управління модулем інтелектуального опрацювання інформації, а також приклади його застосування для обробки запитів надання окремих адміністративних послуг.

У четвертому розділі наведено результати розробки прикладної програмної системи для моделювання та оптимізації процесів у центрі надання адміністративних послуг. Наведено обґрунтування щодо удосконаленої архітектури прикладної програмної системи центрів надання адміністративних послуг для рівня муніципалітетів, яка, на відміну від існуючих містить інтелектуальні модулі обробки інформації про надавані послуги, модулі оптимізації процесів виконання цих послуг за часом виконання і в цілому підвищує ефективність функціонування веб-сервісів. Представлено особливості побудови користувацького інтерфейсу цієї системи.

У завершальній частині цього розділу наведено також результати впровадження розробленої системи для центрів надання адміністративних

послуг та у навчальний процес Західноукраїнського національного університету.

У додатках наведено довідки про впровадження результатів дисертаційного дослідження та фрагменти коду розробленого програмного забезпечення.

Практична цінність роботи полягає у створенні прикладної програмної системи, яка забезпечує підвищення ефективності функціонування веб-сервісів надання адміністративних послуг за рахунок впровадження засобів інтелектуалізації процесу виконання послуг та математичного моделювання функціонування цих сервісів.

Ключові слова: веб-сервіси, архітектура програмного забезпечення, інтелектуальний програмний модуль, центр надання адміністративних послуг, моделювання, ефективність сервісу, інтервальні моделі, ідентифікація параметрів моделі, інформаційні запити, контентна повнота.

ПЕРЕЛІК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковано основні наукові результати дисертації:

1. Dyvak, M., Papa, O., Melnyk, A., Pukas, A., Porplytsya, N., Rot, A. (2020). Interval Model of the Efficiency of the Functioning of Information Web Resources for Services on Ecological Expertise. *Mathematics*, 8(12), 2116. <https://doi.org/10.3390/math8122116>
2. Дивак, М.П., Мельник, А.М., Папа, О.А. (2020). Математичне та програмне забезпечення інтелектуального модуля прикладних програмних систем для надання адміністративних послуг щодо проведення екологічної експертизи. *Інформаційні технології та комп'ютерна інженерія*, 49(3), 66–76. <https://doi.org/10.31649/1999-9941-2020-49-3-66-76>
3. Дивак, М.П., Мельник, А.М., Ковбасистий, А.В., Папа, О.А. (2020). Підхід до математичного моделювання ефективності web-ресурсів. *Оптико-електронні інформаційно-енергетичні технології*, 38(2), 29–37. <https://doi.org/10.31649/1681-7893-2019-38-2-29-37>

Наукові праці, які засвідчують апробацію матеріалів дисертації

4. Kedrin, Y., Dyvak, M., Pukas, A., Voytyuk, I., Maslyiak, Y., Papa, O. (2020). Features of Artificial Bee Colony Based Algorithm Realization for Parametric Identification Method of the Interval Discrete Dynamic Models. 2020 10th International Conference on Advanced Computer Information Technologies (ACIT), 239–245. <https://doi.org/10.1109/ACIT49673.2020.9208916>
5. Pukas, A., Simak, A., Syrnyk, O., Horal, L., Shyjko, V., Papa, O. (2019). Software Module for Data Correctness and Completeness Control in the Academic Staff Performance Appraisal System. 2019 9th International

- Conference on Advanced Computer Information Technologies (ACIT), 277–280. <https://doi.org/10.1109/ACITT.2019.8779999>
6. Пукас А.В., Жук О.О., Кедрін Є.С., Папа О.А. (2019). Інформаційна система підтримки надання послуг ЦНАП. Школа-семінар молодих вчених і студентів. Комп'ютерні інформаційні технології, 29-30.
 7. Papa, O., Kedrin, Y., Pukas, A., Avhustyn, R. (2018). Visitors Queue Management Optimization using Web System for Activity Support of the Administrative Services Center. CEUR Workshop Proceedings, 187-190
 8. Dyvak, M., Maslyiak, Y., Papa, O., Savka, N. (2017). Clustering and interval analysis of heterogeneous data sample. 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), 1, 528–532. <https://doi.org/10.1109/STC-CSIT.2017.8098843>
 9. Кедрін Є.С., Пукас А.В., Папа О.А. (2017). Особливості реалізації веб-орієнтованої системи підтримки діяльності центру надання адміністративних послуг. Сучасні комп'ютерні інформаційні технології. Матеріали I Всеукраїнської конференції з Міжнародною участю ACIT'2017, 168-170.
 10. Кедрін Є.С., Пукас А.В., Веремчук А.В., Папа О.А. (2016). Розширення функціоналу програмної системи підтримки діяльності центру надання адміністративних послуг Тернопільської міської ради. Сучасні комп'ютерні інформаційні технології. Матеріали VI Всеукраїнської школи-семінару молодих вчених та студентів ACIT'2016, 143-144.
 11. Веремчук А.В., Пукас А.В., Порплиця Н.П., Папа О.А. (2016). Програмна система для моделювання об'єктів з розподіленими параметрами на основі інтервальних різницевих операторів. Сучасні комп'ютерні інформаційні технології. Матеріали VI Всеукраїнської школи-семінару молодих вчених та студентів ACIT'2016, 99-101.

12. Пукас А.В., Кедрін Є.С., Папа О.А. (2015). Веб-орієнтована програмна система підтримки діяльності центру надання адміністративних послуг Тернопільської міської ради. Сучасні комп'ютерні інформаційні технології. Матеріали V Всеукраїнської школи-семінару молодих вчених та студентів АСІТ'2015, 94-96.

ANNOTATION

Papa O. A. Modeling the efficiency of web services for administrative services providing - Qualifying scientific work on the rights of the manuscript.

The Ph.D thesis in "Software engineering" (specialty 121) - West Ukrainian National University, Ternopil, 2020.

The qualifications were carried out at the Department of Computer Sciences of the West Ukrainian National University under the Ministry of Education and Science of Ukraine.

The thesis paper is devoted to solving the current scientific and technical problem of developing mathware and software for support of web services of administrative services provision through the introduction of means of intellectualizing the process of performing services and mathematical modelling of the efficiency of web services functioning.

The first section analyses the existing methods and means for organizing web services and administrative services.

The analysis showed that one of the determining factors in the process of creating effective Centres for Provision of Administrative Services and improving the quality of administrative services is the use of modern information technologies. An effective information system for the provision of administrative services will make it possible to obtain objective information on the quality of service provision. In addition, such a system may become the basis for rapid management decisions in order to actually improve and bring the quality of services to the needs of customers. To do this, mainly mobile applications are created that bring a large number of users closer to the services. However, such systems do not reduce the time of service provision, as they require sufficient awareness of the user about the specifics of obtaining certain services, and therefore require intellectualization of the service performance process.

Thus, the task of using intelligent methods of data processing and mathematical methods of evaluating the effectiveness of web services is crucial in order to improve the quality and significantly accelerate the provision of administrative services to citizens as well.

Further in this section the known domestic and foreign software means for administrative services provision have been analysed. The most famous software are the following: iGov - public services portal; diia (action) - public services online; "Service Canada"; "Offices for citizens" - France; "WOM" - Poland.

All of these do not contain intelligent modules or these modules have limited functionality.

Another important issue is to evaluate the effectiveness of web services functioning. The conducted analysis of literature sources showed that mostly performance indicators do not relate to the direct functioning of web resources, which is a significant problem in assessing the quality of web services.

A separate problem is the establishment of a formalized description of the performance indicators of web services, which would allow modelling this performance for different conditions. Thus, the problem of modelling the effectiveness of a web service functioning, which is a component for solving the above mentioned problem, remains relevant due to the variety of these services and the variety of procedures for providing administrative services.

Based on the above mentioned information, the scientific and applied task of developing mathware and software for support of web services providing administrative services through the introduction of means to intellectualize the process of performing services and mathematical modelling of the effectiveness of web services functioning is crucial.

Further in the section the substantiation of means of the online services organization has been carried out. Metrics of efficiency of the online services functioning have been reported.

In the final part of the section the tasks of the thesis research have been set.

The second section of the thesis paper for the first time proposes and substantiates the performance indicator of web-services functioning in the form of user traffic ranking dynamics and its mathematical model, which, unlike existing ones, is based on methods of the interval data analysis and taking into account the cyclical nature of traffic ranking processes, which provided a guaranteed daily forecast of the number of requests to the web service. The developed interval model and the method of its identification provided the possibility of planning and allocation of the organizational resources for the provision of online services. A combination of interval data analysis and parametric model identification methods was used to construct this model. This method, in turn, is based on computational schemes with self-organization by analogy with the behavioural models of the bee colony. Also in this section the technique of obtaining the experimental data from the operating web service has been reported and the example of the interval model identification of this web service efficiency has been considered.

The third section discusses the mathware and software for building an intelligent module in web services for administrative services provision. A new method of intelligent processing of information to assess the similarity of online services, which, unlike existing ones, is based on the clustering of documents that operate on the concepts of distance between objects-details, has been proposed. The proposed method reduces the time of online services provision and generally improves the efficiency of web services.

The method of automated search for optimal sets of documents in a given subject area, represented by a directional graph, which in contrast to the existing uses an ant colony algorithm, which created an intelligent software module, which, in turn, automates the administrative service provision process and, respectively, reduces the provision time for certain categories of services, has been improved.

In the final part of this section the algorithmic software and software of the intelligent module realization as a part of web services has been presented. A scheme for removing duplicate documents and services in the process of forming a search query for administrative services has been developed. The results of the software interface development for management of the intellectual processing of the information module, and also examples of its application for processing of inquiries of separate administrative services provision have been presented.

The fourth section presents the results of the development of an application software system for modelling and optimization of processes in the Centre for Provision of Administrative Services. The substantiation as for the improved architecture of the application software system of the Centres for Provision of Administrative Services for the level of municipalities, which, in contrast to the existing ones, contains intelligent modules for processing information about services provided, modules for optimizing the performance of these services over time and generally increases the efficiency of web services functioning has been reported. The features of construction of the user interface of this system have been presented.

The final part of this section also presents the results of the implementation of the developed system for the Centres for Provision of Administrative Services and in the educational process of the West Ukrainian National University.

The appendices contain information on the implementation of the results of the thesis research and fragments of the code of the developed software.

The practical value of the work is in creation of an application software system that improves the efficiency of web services functioning for the provision of administrative services through the introduction of intellectualization means of the performing services process and mathematical modelling of the functioning of these services.

Keywords: web services, software architecture, intelligent software module, Centre for Provision of Administrative Services, modelling, service efficiency, interval models, model parameter identification, information requests, content completeness.

LIST OF PUBLISHED PAPERS ON THE TOPIC OF THE THESIS PAPER

Scientific works in which the main scientific results of the thesis paper have been published:

1. Dyvak, M., Papa, O., Melnyk, A., Pukas, A., Porplytsya, N., Rot, A. (2020). Interval Model of the Efficiency of the Functioning of Information Web Resources for Services on Ecological Expertise. *Mathematics*, 8(12), 2116. <https://doi.org/10.3390/math8122116>
2. Dyvak, M.; Melnyk, A.; Papa, O. (2020). Mathware and Software of the Intelligent Module of Applied Software Systems for the Provision of Administrative Services as for Conducting Ecological Expertise. *Information technology and computer engineering*, 49(3), 66–76. <https://doi.org/10.31649/1999-9941-2020-49-3-66-76>
3. Dyvak, M. P.; Melnyk, A. M.; Kovbasistyi, A. V.; Papa, O. A. (2020). Approach to Mathematical Modelling of Web Resources Efficiency. *Optoelectronic information and energy technologies*, 38(2), 29–37. <https://doi.org/10.31649/1681-7893-2019-38-2-29-37>

Scientific works that certify the approbation of the thesis paper materials:

4. Kedrin, Y., Dyvak, M., Pukas, A., Voytyuk, I., Maslyiak, Y., Papa, O. (2020). Features of Artificial Bee Colony Based Algorithm Realization for Parametric Identification Method of the Interval Discrete Dynamic Models. 2020 10th International Conference on Advanced Computer Information Technologies (ACIT), 239–245. <https://doi.org/10.1109/ACIT49673.2020.9208916>
5. Pukas, A., Simak, A., Syrnyk, O., Horal, L., Shyjko, V., Papa, O. (2019). Software Module for Data Correctness and Completeness Control in the

- Academic Staff Performance Appraisal System. 2019 9th International Conference on Advanced Computer Information Technologies (ACIT), 277–280. <https://doi.org/10.1109/ACITT.2019.8779999>
6. Pukas A.V., Zhuk O.O., Kedrin Ye.C., Papa O.A. (2019). Information System of Support the Provision of Services by the Centres for Provision of Administrative Services. School-seminar of Young Scientists and Students, Computer Information Technologies, 29-30.
 7. Papa, O., Kedrin, Y., Pukas, A., Avhustyn, R. (2018). Visitors Queue Management Optimization using Web System for Activity Support of the Administrative Services Center. CEUR Workshop Proceedings, 187-190
 8. Dyvak, M., Maslyiak, Y., Papa, O., Savka, N. (2017). Clustering and interval analysis of heterogeneous data sample. 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), 1, 528–532. <https://doi.org/10.1109/STC-CSIT.2017.8098843>
 9. Kedrin Ye.S., Pukas A.V., Papa O.A. (2017). Features of realizing the web-oriented system to support the activities of the Centre for Provision of Administrative Services. Modern computer information technologies. Proceedings of the I All-Ukrainian Conference with International Participation ACIT'2017, 168-170.
 10. Kedrin Ye.S., Pukas A.V., Veremchuk A.V., Papa O.A. (2016). Expansion of the functionality of the software system to support the activities of the Centre for Provision of Administrative Services of Ternopil City Council. Modern computer information technologies. Proceedings of the VI All-Ukrainian school-seminar of young scientists and students ACIT'2016, 143-144.
 11. Veremchuk A.V., Pukas A.V., Porplytsia N.P., Papa O.A. (2016). Software system for modelling objects with distributed parameters based on interval difference operators. Modern computer information technologies. Proceedings

of the VI All-Ukrainian school-seminar of young scientists and students ACIT'2016, 99-101.

12. Pukas A.V., Kedrin Ye.S., Papa O.A. (2015). Web-oriented software system to support the activities of the Centre for Provision of Administrative Services of Ternopil City Council. Modern computer information technologies. Proceedings of the V All-Ukrainian school-seminar of young scientists and students ACIT'2015, 94-96.

ЗМІСТ

АНОТАЦІЯ	2
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	20
ВСТУП.....	21
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА ЗАСОБІВ ОРГАНІЗАЦІЇ ВЕБ-СЕРВІСІВ ТА АДМІНІСТРАТИВНИХ ПОСЛУГ	30
1.1. Огляд сучасних систем надання онлайн послуг та перспективи розвитку	30
1.2. Програмні та технічні засоби організації онлайн сервісів.....	42
1.3. Аналіз ефективності функціонування онлайн сервісів	54
1.4. Постановка задачі дисертаційного дослідження.....	58
Висновки до розділу 1	59
РОЗДІЛ 2 ІНТЕРВАЛЬНА МОДЕЛЬ ЕФЕКТИВНОСТІ ФУНКЦІОНУВАННЯ ІНФОРМАЦІЙНИХ ВЕБ-РЕСУРСІВ.....	61
2.1. Обґрунтування показника ефективності функціонування інформаційних веб-ресурсів.....	62
2.2. Інтервальна модель ефективності функціонування інформаційних веб-ресурсів та метод її ідентифікації	65
2.3. Методика отримання експериментальних даних	74
2.4. Приклад ідентифікації інтервальної моделі ефективності функціонування інформаційних веб-ресурсів.....	76
Висновки до розділу 2	82
РОЗДІЛ 3 МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ПРИКЛАДНИХ ПРОГРАМНИХ СИСТЕМ ДЛЯ НАДАННЯ АДМІНІСТРАТИВНИХ ПОСЛУГ	83
3.1 Математичне забезпечення інтелектуального модуля.....	84
3.2. Алгоритмічне забезпечення реалізації інтелектуального модуля.....	90
3.3. Особливості реалізації програмного забезпечення	93
Висновки до розділу 3	104

РОЗДІЛ 4 ПРИКЛАДНА ПРОГРАМНА СИСТЕМА ДЛЯ МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЇ ПРОЦЕСІВ ДЛЯ ЦЕНТРУ НАДАННЯ АДМІНІСТРАТИВНИХ ПОСЛУГ	106
4.1. Архітектура програмної системи	106
4.2. Прикладні аспекти реалізації системи для центру надання адміністративних послуг	116
4.3. Особливості організації користувацького інтерфейсу.....	130
Висновки до розділу 4	150
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	153
ДОДАТОК А	167
ДОДАТОК Б.....	187
ДОДАТОК В.....	189

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ЦНАП - центр надання адміністративних послуг.

АБК - алгоритм бджолоїної колонії.

SOA - сервісно-орієнтована архітектура.

XML - розширювана мова розмітки.

WSDL - мова визначення інтерфейсу вебсервісу.

SOAP - протокол обміну структурованими повідомленнями.

UDDI - платформово-незалежний інструмент.

HTML - мова гіпертекстової розмітки.

HTTP - протокол передачі даних.

PHP - скриптова мова програмування.

API - прикладний програмний інтерфейс.

ВСТУП

Актуальність теми. Створення цифрових систем надання адміністративних послуг сьогодні є одним із основних світових трендів. Останнім часом тільки в Україні було створено понад тисячу центрів веб-сервісів надання послуг такого характеру.

Одним із визначальних чинників процесу створення ефективних ЦНАП та підвищення якості надання адміністративних послуг є використання сучасних інформаційних технологій. Ефективна інформаційна система надання адміністративних послуг дасть можливість одержувати об'єктивну інформацію щодо їх якості. Така система може стати основою для оперативного ухвалення управлінських рішень, значно підвищити рівень якості послуг та наблизити їх до потреб замовників. Для цього переважно створюються мобільні додатки, які наближають до сервісів велику кількість користувачів. Водночас такі системи не зменшують часу надання послуг, оскільки вимагають достатньої обізнаності користувача з особливостями отримання тих чи інших послуг, а отже, потребують інтелектуалізації процесу виконання послуги. Таким чином, актуальною є задача використання інтелектуальних методів обробки даних та математичних методів оцінювання ефективності веб-сервісів для підвищення якості та значного прискорення процесу надання адміністративних послуг для громадян.

Найвідомішими вітчизняними та зарубіжними програмними засобами на базі для надання адміністративних послуг є: iGov – портал державних послуг; дія – державні послуги онлайн; “Сервіс Канада”; “Офіси для громадян” – Франція; “WOM” – Польща. Усі із перерахованих не містять інтелектуальних модулів або ці модулі мають обмежений функціонал.

Питаннями створення та підвищення ефективності функціонування веб-сервісів займалися вітчизняні та зарубіжні вчені. Серед них слід виділити

роботи таких науковців як: А. Пелешишина, В. Литвина, В. Дубового R.Kosala, F. Menczer, A. Fox, P. Czarnul, J. Bicevskis.

Однак, у дослідженнях цих вчених відсутні методи інтелектуалізації надання адміністративних послуг, а саме питання ефективності їх функціонування не є чітко означеним.

Окремою проблемою є встановлення формалізованого опису показників ефективності веб-сервісів, який би давав можливість моделювання цієї ефективності у різних умовах. Задачі оцінювання ефективності веб-сервісів розглянуто в працях таких вітчизняних та зарубіжних вчених: В. Захарова, А. Дергачев, О. Беседовський, Т. Yu, Z. Jeff.

Проте, проблема моделювання ефективності функціонування веб-сервісу, яка є складовою для розв'язування вище зазначеної задачі, залишається актуальною в силу різноманіття цих сервісів та різноманіття процедур надання адміністративних послуг.

Виходячи із вищезазначеного, актуальною є науково-прикладна задача розробки математичного та програмного забезпечення підтримки веб-сервісів надання адміністративних послуг шляхом впровадження засобів інтелектуалізації процесу виконання послуг та математичного моделювання ефективності функціонування веб-сервісів.

Створення зазначених засобів створює можливості для розробки ряду прикладних програмних систем для виявлення неактуальної та некоректної інформації на інформаційних веб-ресурсах.

Мета і завдання дослідження. Метою дисертаційного дослідження є підвищення ефективності функціонування веб-сервісів надання адміністративних послуг шляхом впровадження засобів інтелектуалізації процесу виконання послуг та математичного моделювання ефективності функціонування веб-сервісів.

Для досягнення цієї мети необхідно виконати такі завдання:

- проаналізувати існуючі методи та засоби організації веб-сервісів для надання адміністративних послуг;
- обґрунтувати показник оцінки ефективності функціонування web-сервісів та побудувати метод його ідентифікації і моделювання;
- розробити метод інтелектуального опрацювання інформації для оцінювання подібності онлайн послуг;
- удосконалити метод автоматизованого пошуку оптимальних наборів документів в заданій предметній області з інтелектуальним модулем;
- розробити архітектуру прикладної програмної системи центрів надання адміністративних послуг на рівні муніципалітетів, яка містить інтелектуальні модулі обробки інформації про надавані послуги та модулі оптимізації процесів виконання цих послуг за часом виконання;
- провести апробацію запропонованих методів та архітектури прикладної програмної системи при вирішенні прикладних задач щодо підвищення ефективності функціонування веб-сервісів надання адміністративних послуг.

Об'єкт дослідження – функціонування веб-сервісів надання адміністративних послуг.

Предмет дослідження – математичне та програмне забезпечення підтримки функціонування веб-сервісів надання адміністративних послуг.

Методи дослідження. В процесі розробки математичної моделі ефективності функціонування web-сервісів у вигляді динаміки відвідуваності користувачами використано теоретико-множинний підхід та методи аналізу інтервальних даних. Для розробки методу інтелектуального опрацювання інформації щодо оцінювання подібності онлайн послуг – методи комп'ютерної лінгвістики та семантичного опису. З метою розробки методу автоматизованого пошуку оптимальних наборів документів в заданій предметній області, представлених напрямленим графом, використано теорію графів та метод оптимізації на основі поведінкових алгоритмів

штучної мурашиної колонії. Розвиток та удосконалення архітектури створеного програмного забезпечення здійснено із використанням методів об'єктно-орієнтованого аналізу та проектування. Для проектування та опису програмного забезпечення використано середовище UML-2.0

Наукова новизна отриманих результатів. У межах дисертаційної роботи вперше:

- запропоновано та обґрунтовано математичну модель ефективності функціонування web-сервісів у вигляді динаміки відвідуваності користувачами, яка, на відміну від існуючих, побудована на методах аналізу інтервальних даних та урахування циклічності процесів відвідуваності, що забезпечує гарантований прогноз кількості запитів веб-сервісу, можливість планування та розподілу ресурсів організації щодо надання онлайн послуг і підвищення ефективності функціонування веб-сервісу загалом;
- запропоновано та обґрунтовано метод інтелектуального опрацювання інформації для оцінювання подібності онлайн послуг, який, на відміну від існуючих, ґрунтується на кластеризації документів, що оперують поняттями відстані між об'єктами-реквізитами. Запропонований метод забезпечує зниження часу надання онлайн послуг і у цілому забезпечує підвищення ефективності веб-сервісів.

Набули подальшого розвитку:

- метод автоматизованого пошуку оптимальних наборів документів в заданій предметній області, представлених напрямленим графом, який, на відміну від існуючих, використовує алгоритм мурашиної колонії, що забезпечило створення інтелектуального програмного модуля з автоматизацією процесу надання адміністративної послуги;
- архітектура прикладної програмної системи центрів надання адміністративних послуг на рівні муніципалітетів, яка, на відміну від існуючих, містить інтелектуальні модулі обробки інформації про послуги, що

надаються, оптимізації процесів їх виконання за часом і в цілому підвищує ефективність функціонування веб-сервісів.

Особистий внесок здобувача. Усі результати, викладені в дисертаційній роботі, отримані автором самостійно. У друкованих працях, опублікованих у співавторстві, автор отримав такі результати:

- [1] створено інтервальну модель ефективності функціонування інформаційних веб-ресурсів для надання адміністративних послуг;
- [2] запропоновано математичне та програмне забезпечення інтелектуального модуля прикладних програмних систем для надання адміністративних послуг;
- [3] запропоновано підхід до математичного моделювання ефективності веб-ресурсів;
- [4] описано особливості реалізації алгоритму бджолоїної колонії для методу параметричної ідентифікації;
- [5] запропоновано оцінку моделювання ефективності веб-системи;
- [6] запропоновано оптимізацію веб-системи для підтримки діяльності центру адміністративних послуг ;
- [7] створено інформаційну систему підтримки надання послуг ЦНАП;
- [8] запропоновано кластеризацію та інтервальний аналіз вибірки даних;
- [9] запропоновано реалізацію веб-орієнтованої системи підтримки діяльності центру надання адміністративних послуг;
- [10] удосконалено функціонал програмної системи підтримки діяльності центру надання адміністративних послуг Тернопільської міської ради;
- [11] створено програмну систему для моделювання об'єктів з розподіленими параметрами на основі інтервальних різницевих операторів;

- [12] створено веб-орієнтовану програмну систему підтримки діяльності центру надання адміністративних послуг.

Основні положення та результати дисертаційної роботи в наведених працях викладені в повному обсязі.

Апробація результатів дисертації. Основні положення і результати дисертаційної роботи доповідались та обговорювались на семи науково-технічних конференціях в Україні й за кордоном та багатьох наукових семінарах, а саме:

- 2020 10th International Conference on Advanced Computer Information Technologies (ACIT), Deggendorf, Germany, 2020;
- 2019 9th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech Republic, 2019;
- 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), Lviv, 2017;
- “Сучасні комп’ютерні інформаційні технології” Матеріали I Всеукраїнської конференції з Міжнародною участю ACIT'2017, 2017;
- “Сучасні комп’ютерні інформаційні технології” Матеріали VI Всеукраїнської школи-семінару молодих вчених та студентів ACIT'2016, 2016;
- “Сучасні комп’ютерні інформаційні технології” Матеріали V Всеукраїнської школи-семінару молодих вчених та студентів ACIT'2015, 2015;
- Школа-семінар молодих вчених і студентів, Комп’ютерні інформаційні технології, м.Тернопіль, 2019;
- Наукові семінари кафедри комп’ютерних наук ЗУНУ (2016-2020 рр.).

Публікації.

За результатами роботи опубліковано 12 наукових праць: 2 статті у фахових наукових виданнях України та одна стаття у закордонному фаховому виданні, що входить до міжнародних наукометричних баз Scopus та Web of Science, 9 публікацій у матеріалах конференцій, 4 з яких входять до міжнародних наукометричних баз Scopus та Web of Science.

Структура та обсяг роботи. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 131 найменувань та 3 додатків. Загальний обсяг роботи складає 194 сторінок друкованого тексту, з них 132 сторінка основного тексту. Робота містить 77 рисунків і 13 таблиць.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційна робота виконувалася в межах пріоритетного напрямку розвитку науки і техніки «Інформаційні та комунікаційні технології», визначеного Законом України «Про пріоритетні напрями розвитку науки і техніки (№2623-III від 11.07.2001 р. в редакції від 16.01.2016 р.), а також згідно з планом науково-дослідних робіт кафедри комп'ютерних наук Тернопільського національного економічного університету протягом 2016 – 2019 років. Основні результати дисертаційного дослідження отримано в межах виконання таких тем:

- держбюджетної науково-технічної (експериментальної) розробки молодих вчених: «Математичне та програмне забезпечення для контролю забруднення атмосфери автотранспортом» (державний реєстраційний номер 0116U005507);
- госпдоговірних науково-дослідних робіт: «Веб-орієнтована програмна система підтримки діяльності Центру надання адміністративних послуг Тернопільської міської ради» (державний реєстраційний номер 0115U004825), «Розширення функціоналу програмної системи

підтримки діяльності Центру надання адміністративних послуг Тернопільської міської ради» (державний реєстраційний номер 0116U008541), «Модернізація програмної системи підтримки діяльності Центру надання адміністративних послуг Тернопільської міської ради» (державний реєстраційний номер 0117U003870).

Автором розроблено прикладну програмну систему, яка забезпечує підвищення ефективності функціонування веб-сервісів надання адміністративних послуг за рахунок впровадження засобів інтелектуалізації процесу виконання послуг та математичного моделювання ефективності функціонування цих сервісів.

Усі вищезгадані роботи виконувалися за безпосередньої участі автора.

Практичне значення отриманих результатів полягає у створенні прикладної програмної системи, яка забезпечує підвищення ефективності функціонування веб-сервісів надання адміністративних послуг за рахунок впровадження засобів інтелектуалізації процесу виконання послуг та математичного моделювання ефективності функціонування даних сервісів.

Розроблену програмну систему апробовано при створенні веб-сервісів прикладної програмної системи ЦНАП Тернопільської міської ради, а також реалізовано для оцінювання ефективності цих сервісів. Застосування розробленого програмного забезпечення та математичної моделі показника ефективності веб-сервісів дало можливість сформулювати вимоги для SEO прикладної програмної системи ЦНАП Тернопільської міської ради, зокрема, щодо налаштування параметрів продуктивності, зменшення впливу структурованості контенту на продуктивність ресурсу та оцінювання ефективності його функціонування під навантаженням.

Теоретичні та прикладні результати дисертаційної роботи використано:

- в центрі надання адміністративних послуг Тернопільської міської ради забезпечує підвищення ефективності функціонування веб-сервісів

- надання адміністративних послуг за рахунок впровадження засобів інтелектуалізації процесу виконання послуг та математичного моделювання ефективності функціонування цих сервісів (акт про впровадження результатів дисертаційної роботи від 22 жовтня 2020 р.);
- в навчально-науковому центрі інформаційних технологій Західноукраїнського національного університету забезпечує підвищення ефективності комунікації із зовнішніми додатками за рахунок скорочення часу надання послуг, а використання математичної моделі оцінки ефективності функціонування web-сервісів у вигляді динаміки відвідуваності користувачами дозволило спрогнозувати гарантований щоденний прогноз кількості запитів веб-сервісу (акт про впровадження результатів дисертаційної роботи від 16 жовтня 2020 р.);
 - у Тернопільській міській громадській організації «Тернопільський освітній комунікаційний центр» (акт про впровадження результатів дисертаційної роботи від 30 жовтня 2020 р.);
 - у навчальному процесі Західноукраїнського національного університету на кафедрі комп'ютерних наук при викладанні дисциплін «Архітектура та проектування програмного забезпечення», «Конструювання програмного забезпечення» та «Аналіз вимог до програмного забезпечення» розглядаються методи побудови архітектури прикладної програмної системи, засоби її розробки у випадку реалізації інтелектуальних програмних модулів, створених на основі поведінкової моделі мурашиної колонії. При викладанні дисципліни «Якість програмного забезпечення та тестування» у частині метрики та атрибути якості розглянуто методи дослідження ефективності веб-ресурсів та моделювання показників якості на основі інтервального аналізу даних. (акт про впровадження результатів дисертаційної роботи від 27 жовтня 2020 р.).

РОЗДІЛ 1

АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА ЗАСОБІВ ОРГАНІЗАЦІЇ ВЕБ-СЕРВІСІВ ТА АДМІНІСТРАТИВНИХ ПОСЛУГ

Одним із визначальних чинників процесу створення ефективних ЦНАП та підвищення якості надання адміністративних послуг є використання сучасних інформаційних технологій [13]. Ефективна інформаційна система надання адміністративних послуг дасть можливість одержувати об'єктивну інформацію щодо їх якості. Використання інтелектуальних методів обробки даних може стати основою для оперативного ухвалення управлінських рішень, значно покращити рівень якості послуг, наблизити їх до потреб замовників, прискорити надання адміністративних послуг для громадян [14].

У даному розділі розглянуто особливості функціонування сучасних інформаційних систем надання адміністративних послуг. Обґрунтовано вибір інструментальних засобів розробки програмного забезпечення, проаналізовано основні показники функціонування веб-додатків центрів надання адміністративних послуг. Розкрито актуальність науково-практичної задачі розробки математичного та програмного забезпечення підтримки веб-сервісів надання адміністративних послуг шляхом впровадження засобів інтелектуалізації процесу виконання послуг і математичного моделювання ефективності функціонування веб-сервісів.

1.1. Огляд сучасних систем надання онлайн послуг та перспективи розвитку

«Сервіс Канада» був створений урядом Канади у вересні 2005 року, щоб забезпечити канадцям простіший, універсальний доступ до програм та послуг, які їм потрібні через Інтернет [15].

«Сервіс Канада» загалом має понад 630 точок обслуговування та 20 тисяч працівників. Середній показник відвідуваності сайту за день складає понад 183 тисячі. На рисунку 1.1 наведено головну сторінку «Сервісу Канада».



Рисунок 1.1. Головна сторінка «Сервісу Канада».

Серед найпопулярніших послуг «Сервіс Канада»: призначення допомоги у випадку безробіття; надання номеру соціального страхування; видача паспорта; послуги у сфері пенсійного забезпечення. Офіси «Сервіс Канада» можуть надавати не лише свої послуги, а й ті, що загалом належать до інших юрисдикцій [15].

В даному сервісі усі послуги представлені в алфавітному порядку.

Дана організація використовує сегментацію клієнтів: за віком, за соціально-економічним статусом тощо. Для кожного сегменту розробляється своя стратегія надання послуг (бажаний спосіб отримання послуг, інформації тощо).

“Офіс для громадян” – Франція (service-public.fr) був створений у 2000 році. Поточна версія веб-ресурсу повністю розроблена в адаптивному дизайні. Це частина глобального проекту, покликаного запропонувати користувачам єдину точку входу для отримання інформації та онлайн-послуг.

Щомісяця даний веб-ресурс відвідують понад 4 млн. користувачів. Портал надає доступ до усієї інформації у сфері адміністративних послуг, структурованої чітко і просто за секціями [16].

На даному ресурсі є чітко визначені три основні групи послуг за суб'єктами звернення: послуги для громадян (Particuliers), для бізнесу (Professionnels – підприємців та юридичних осіб), для об'єднань громадян (Associations).

Основними розділами веб-ресурсу є: новини, послуги та форми, служба обміну повідомленнями та запитання.

Головна сторінка порталу зображена на рис.1.2. Вона містить детальнішу рубрикацію послуг за сферами суспільних відносин: питання іноземців і громадянства; сім'ї; трудових відносин; правосуддя; нерухомості; охорони здоров'я та соціальних послуг тощо.

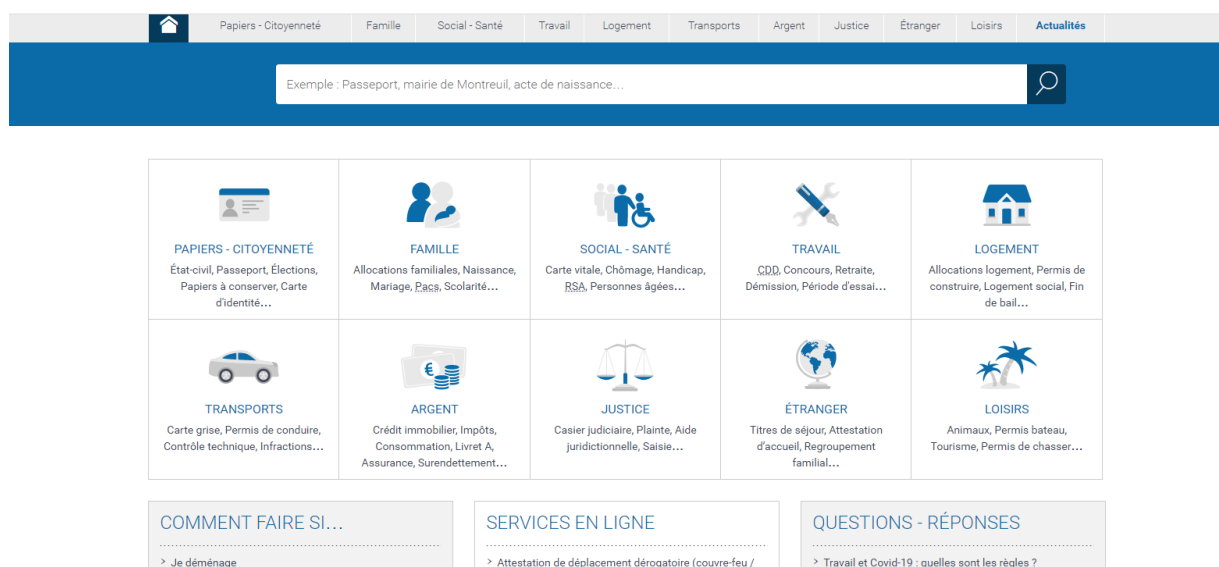


Рисунок 1.2. Головна сторінка “Офіс для громадян”, Франція

Однією з особливостей веб-ресурсу є відображення статистики відвідувань, яка зображена на рисунку 1.3.

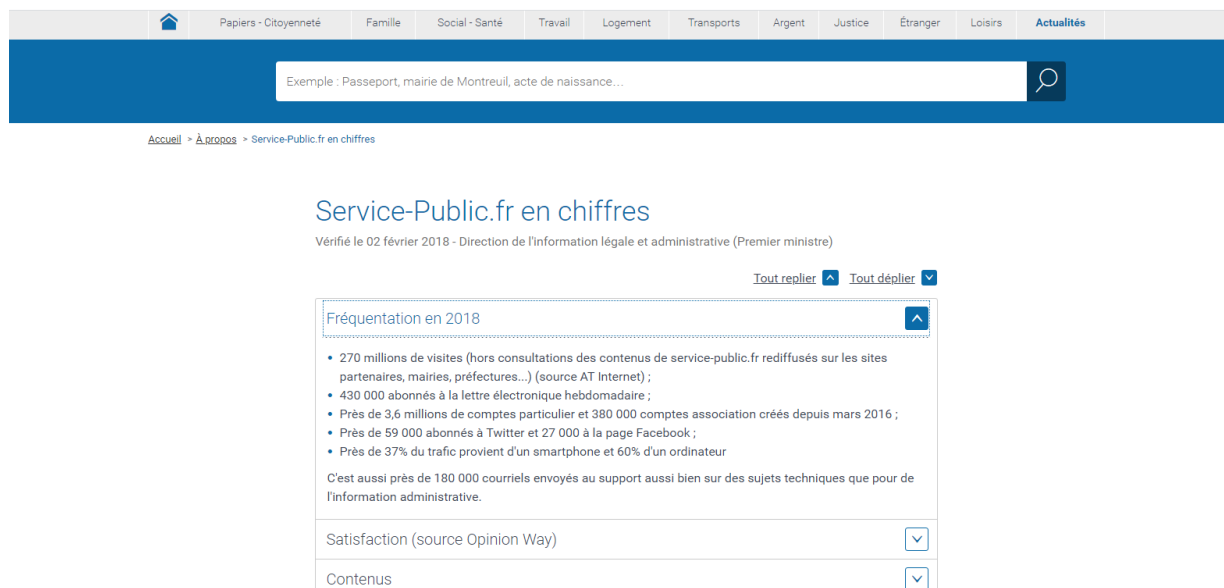


Рисунок 1.3. Статистики відвідувань “Офісу для громадян”, Франція

Портал Республіки Польща є основною точкою доступу до публічної інформації через польську адміністрацію. Інформаційна архітектура порталу базуватиметься на трьох групах: послуги для громадян, для бізнесу, для об’єднань громадян (рис. 1.4) [17].

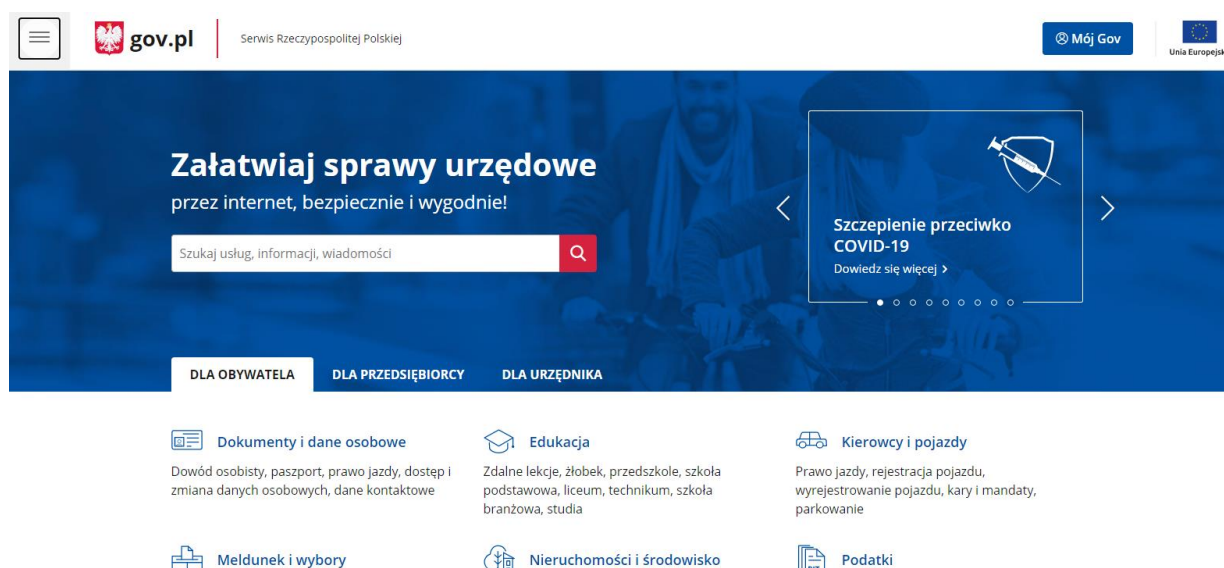


Рисунок 1.4. Інформаційна архітектура порталу Республіки Польща

При наявному акаунту на порталі громадянин може безпечно спілкуватися з адміністрацією. Варто відзначити упорядкованість офіційних документів за галузевим критерієм. Щодо кожної послуги існує детальна інформація: номер картки (бланк заяви, яку необхідно заповнити); перелік необхідних до подання документів; місце подання документів та отримання рішення у справі, порядок оплати послуг; термін надання споживачам відповіді тощо [17].

Портал державних послуг «Дія» – український е-сервіс державних послуг, що розробляється Міністерством цифрової трансформації України [18]. Це мобільний застосунок із цифровими документами та портал із публічними послугами, створений в 2020 році. На рисунку 1.5 зображено головну сторінку portalу.

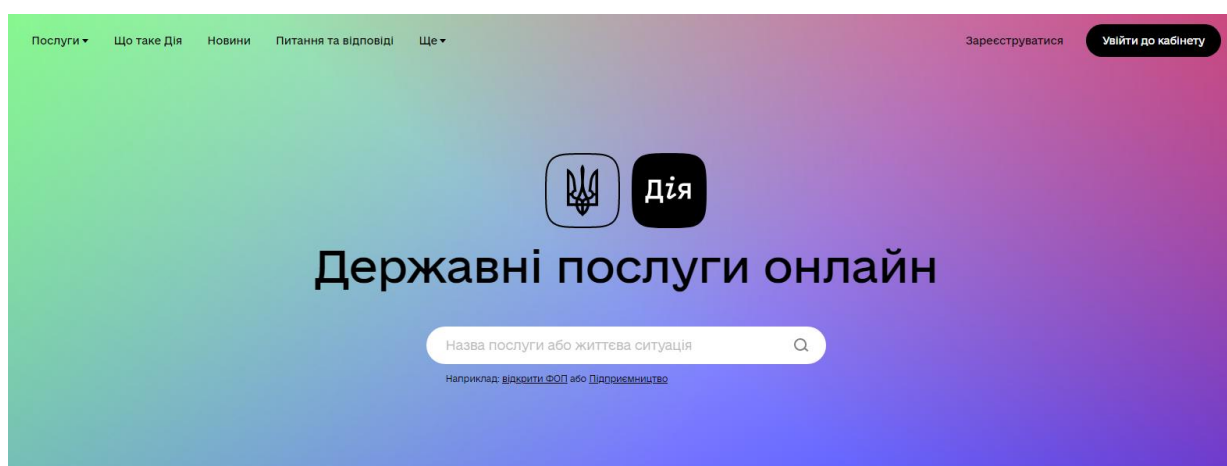


Рисунок 1.5. Головна сторінка portalу Дія.

Дія – держава в смартфоні, реалізована за допомогою мови програмування: JavaScript/React. Основні можливості portalу – використання цифрових документів та отримання державних послуг [18].

Усі послуги поділені на категорії. На сторінці послуги є її опис, зазначено терміни виконання, результат, вартість, розписані правила отримання послуги та перелік необхідних документів (рис. 1.6).

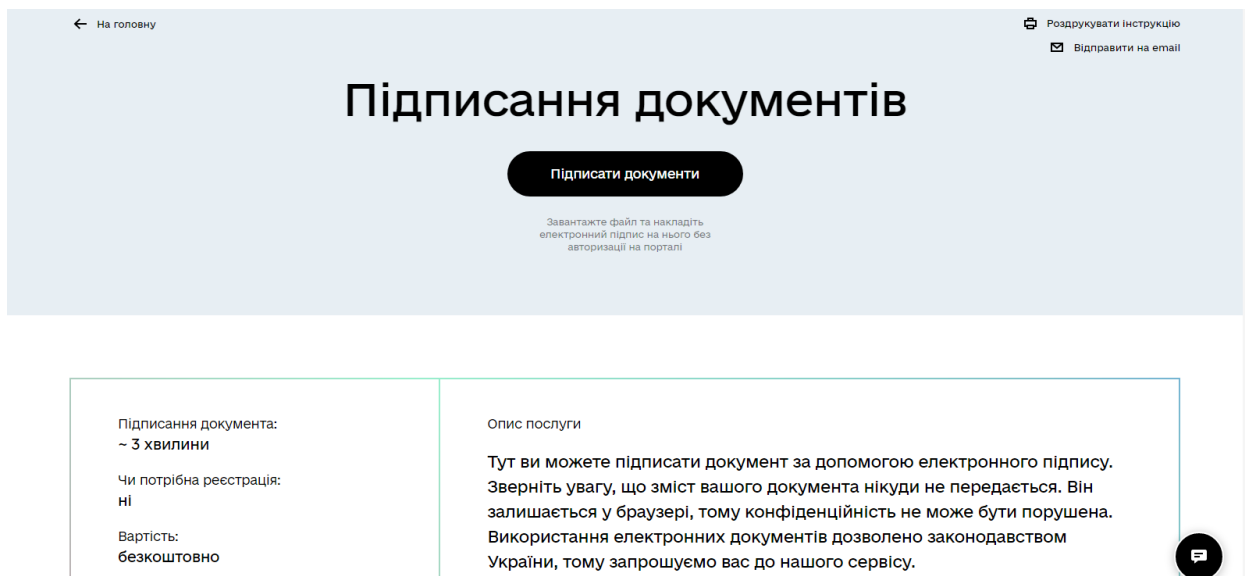


Рисунок 1.6. Сторінка інформації про послугу portalу Дія

Особливості portalу: авторизація через систему BankID НБУ, онлайн та автономний режим, перевірка цифрових документів через тимчасові унікальні коди, автоматична верифікація РНОКПП, відображення техпаспорта у застосунку належного користувача, сканер QR-кодів, повідомлення про штрафи, виконавчі провадження та важливі новини, онлайн-підтримка та чатботи [18-22].

iGov – це волонтерський проект. Цей портал зроблено волонтерською командою iGov в рамках боротьби з корупцією в Україні та вдосконалення бізнес-процесів в державних органах. В ньому зібрано послуги, які державні органи України надають громадянам та бізнесу [19].

На рис. 1.7 зображено головну сторінку проекту iGov.

Послуги розділяються на дві категорії: послуги для громадян та послуги для бізнесу. Кожна послуга містить наступну інформацію: загальну інформацію про послугу, порядок виконання, перелік необхідних документів, терміни виконання.

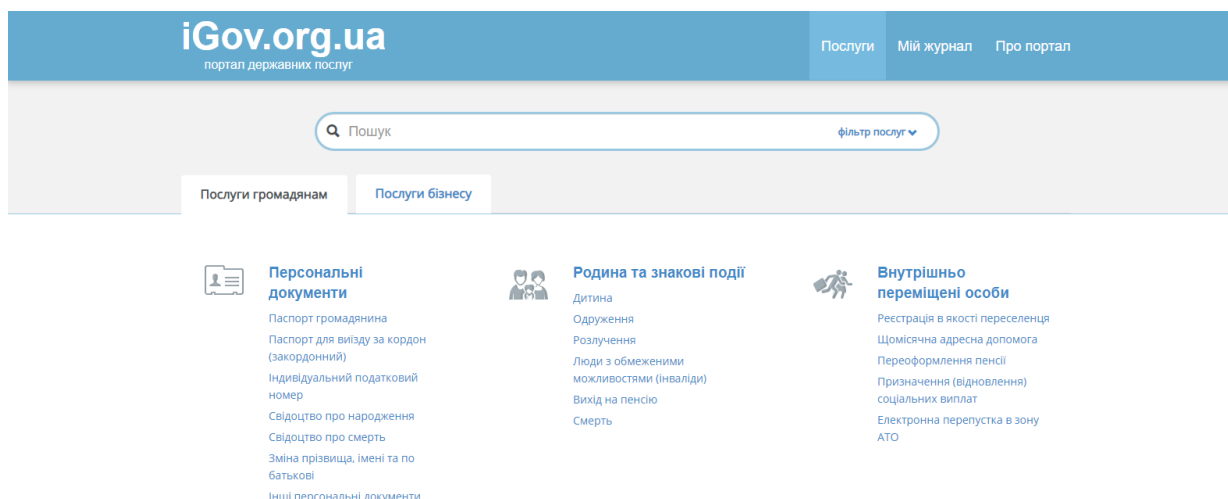


Рисунок 1.7. Головна сторінка порталу iGov

Реалізований пошук за ключовими словами, який спрощує завдання, дає можливість отримати список доступних послуг щодо певного регіону. Також на даному порталі можна переглянути статистику виконання тієї чи іншої послуги.

Сьогодні день існує багато сайтів центрів надання послуг, які мають обласний або загальнодержавний характер. Кожен із них має схожий функціонал і спрямований на вирішення питань громадян. Разом з тим, кожен із них має свої особливості як в підходах, так і в використовуваних засобах та технологіях.

Центр надання адміністративних послуг міста Івано-Франківська (www.cnap.if.ua).

Головна сторінка сайту зображена на рисунку 1.8. Вгорі сторінки можна побачити контактну інформацію, а також завантажити додатки для мобільних пристроїв щодо роботи з послугами центру. Нижче представлена форма входу в персональний кабінет, дана карта розміщення, а також віртуальна екскурсія центром. Одразу на головній сторінці подані категорії послуг з можливістю переключити вигляд, перелік та сортування. Для

наглядності у правій колонці подана інфографіка із числом наданих послуг, консультацій, кількість обслужених осіб [20].

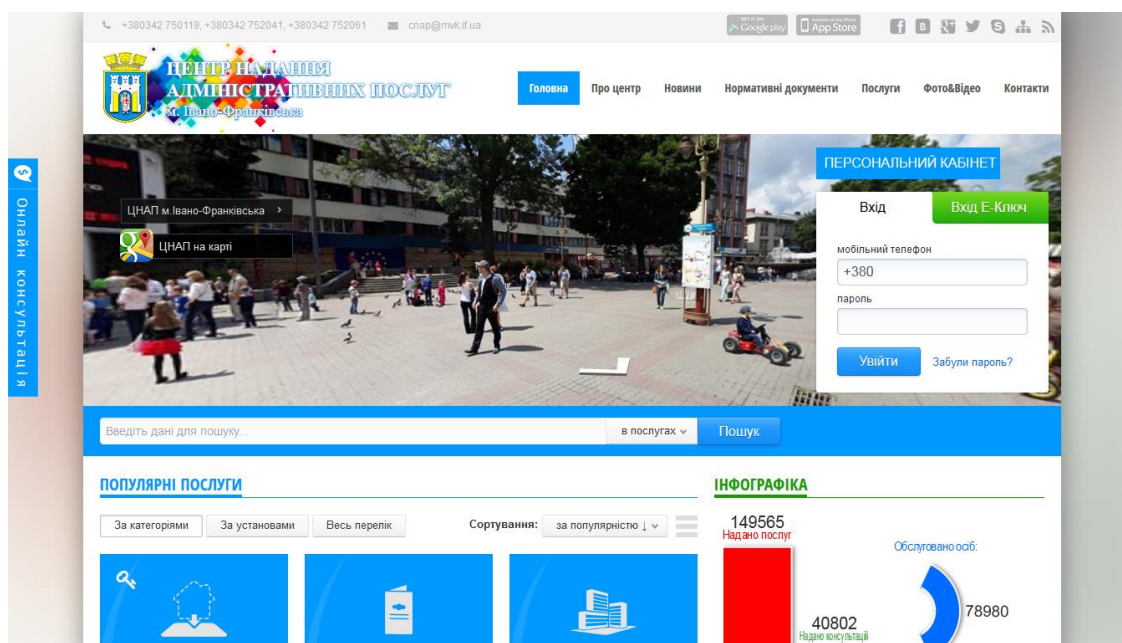


Рисунок 1.8. Головна сторінка сайту (snar.if.ua)

Для того, щоб записатись на прийом (рис. 1.9), потрібно обрати дату, підрозділ, бажаний час. Після цього потрібно ввести свої персональні дані для ідентифікації і завершити реєстрацію.

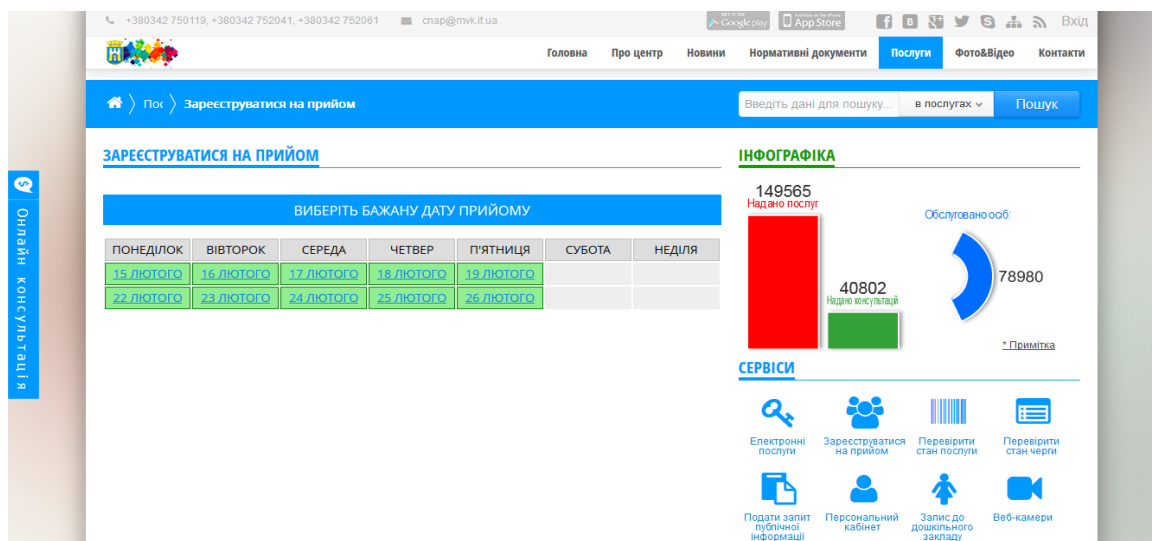


Рисунок 1.9. Сторінка запису на прийом (snar.if.ua)

На рисунку 1.10 наведений приклад сторінки з інформацією про послугу. Окрім основної інформації тут є можливість скачати інформаційну картку з детальними даними та лишити коментар [20].

81/05 Земельні питання. (Відділ Держземагентства)
Надання довідки з державної статистичної звітності про наявність земель та розподіл їх за власниками земель, землекористувачами, угіддями (за даними форми 6-зем)

Термін виконання	14
Виконавець	Озарко Л.Я.
Результат послуги	Довідка з державної статистичної звітності про наявність земель та розподіл їх за власниками земель, землекористувачами, угіддями (за даними форми 6-зем)
Перелік документів	1. Заява про надання відомостей з Державного земельного кадастру за визначеною формою. 2. Оригінал документації із землеустрою. 3. Правоустановчі документи на земельну ділянку землевласника/землекористувача (за наявності). 4. Копія паспорта, довідки про присвоєння ідентифікаційного номера (для фізичних осіб) або реєстраційні документи суб'єкта господарської діяльності (для юридичних осіб).
Вкладення	Інформаційна картка

Послугу надано: 2849

ІНФОГРАФІКА

149565 Надано послуг
40802 Надано консультацій
Обслуговано осіб: 78980

СЕРВІСИ

- Електронні послуги
- Зареєструватися на прийом
- Перевірити стан послуги
- Перевірити стан черги
- Подати запит публічної інформації
- Персональний кабінет
- Залис до дошкільного закладу
- Веб-камери

ОСТАННІ НОВИНИ

Відтепер ви самі обираєте місце реєстрації бізнесу та нерухомості за якістю сервісу та зручністю розташування.

Рисунок 1.10. Сторінка інформації про послугу (спар.if.ua)

Портал адміністративних послуг міста Луцька (www.ap.volyn.ua).

На головній сторінці сайту (рис. 1.11) розміщено основне меню з 5-ма пунктами. Нижче знаходяться пункти, які допомагають знайти відвідувачу потрібну йому послугу. Для зручності послуги розміщуються в різних групах (за сферами надання, за життєвими ситуаціями, за суб'єктами надання, популярні послуги, онлайн-послуги). Сайт доступний на двох мовах – українська і англійська [21].



Рисунок 1.11. Головна сторінка сайту (ar.volyn.ua)

Вигляд сторінки з інформацією про послугу показаний на рис. 1.12. Якщо послуга може надаватись в онлайн-режимі, для клієнта з'являється кнопка «Замовити послугу». Це дає змогу подати заявку на отримання послуги прямо з дому, що спрощує взаємодію із центром і є великою перевагою для клієнта [21].

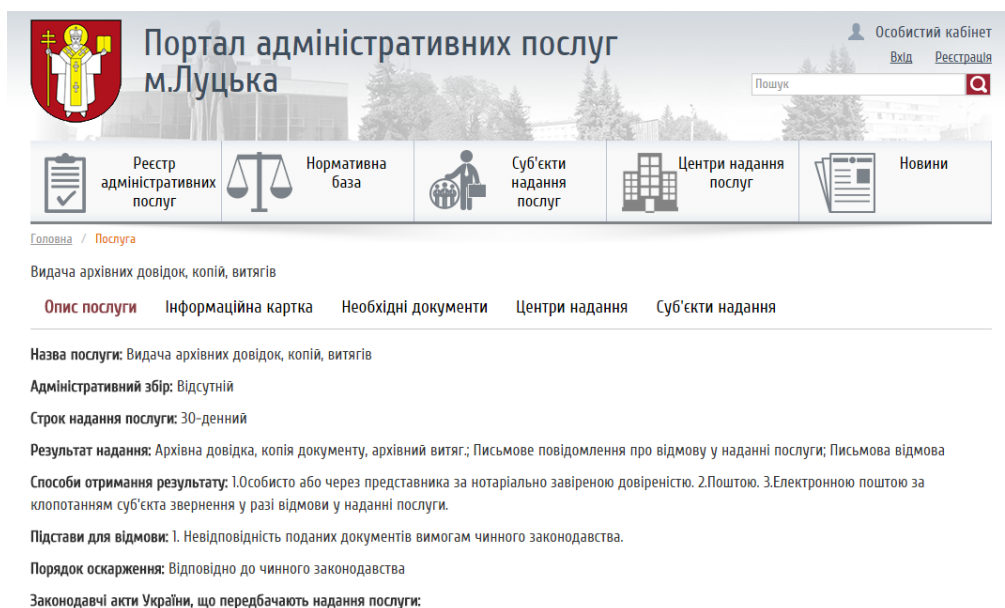


Рисунок 1.12. Сторінка інформації про послугу (ar.volyn.ua)

Для того, щоб отримати онлайн-послугу, клієнту необхідно авторизуватись на сайті. Після авторизації буде виведене вікно з кроками замовлення онлайн-послуги (рис. 1.13). Для продовження необхідно обрати центр надання послуг і натиснути кнопку «Далі». Після вибору центру надання послуг необхідно завантажити необхідні документи. Далі потрібно провести ідентифікацію користувача за допомогою Bank ID або ЕЦП. Після ідентифікації замовлення клієнт отримає повідомлення про прийняття заявки [21].

Рисунок 1.13. Процес замовлення онлайн-послуги (ар.volyn.ua)

Рівненський регіональний центр надання адміністративних послуг (www.cnar.vv.ua).

Як і на більшості сайтів центрів, на головній сторінці розміщені послуги, які можна переглядати за сферами або за суб'єктами надання (рис. 1.14) [22].

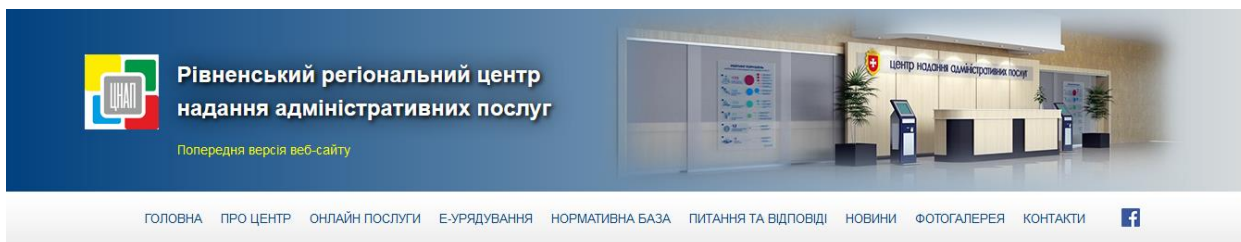


Рисунок 1.14. Головна сторінка сайту (спар.rv.ua)

На рисунку 1.15 наведений приклад сторінки з інформацією про послугу.

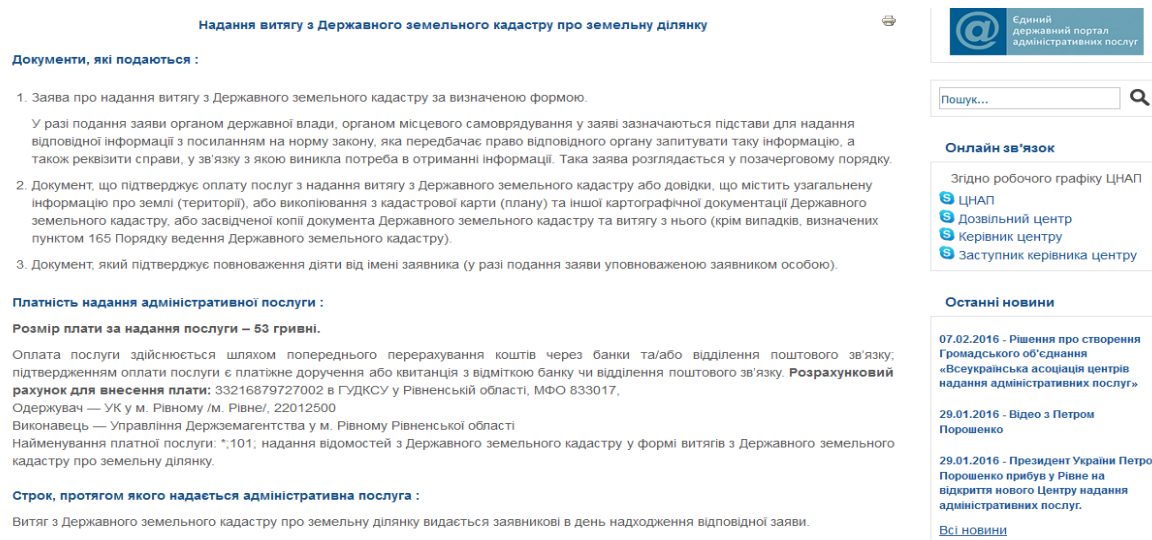


Рисунок 1.15. Сторінка інформації про послугу (спар.rv.ua)

Порівняльна характеристика даних систем подана у таблиці 1.1.

Таблиця 1.1

Порівняння систем

Критерії	Дія	iGov	Портал Республіки Польща	“Офіс для громадян” - Франція	“Сервіс Канада”	ЦНАП м. Івано-Франківськ
Адреса сайту	diia.gov.ua	igov.org.ua	gov.pl	service-public.fr	canada.ca	снар.if.ua
Надання електронних послуг	+	+	+	+	+	+
Запис в електронну чергу	+	+	+	+	+	+
Наявність статистики	+	+	+	+	+	+
Інтелектуальний аналіз даних	+/-	-	+/-	+/-	-	-
Наявність інтелектуальних модулів	-	-	-	-	+/-	-
Допомога користувачу	Онлайн конс.	Онлайн конс.	Онлайн конс.	Онлайн конс.	Онлайн конс.	Онлайн конс.
Мобільна версія	+	+	+	+	+	/-
Мови	Укр.	Укр.	Пол. і англ.	Франц. І англ.	Франц. І англ.	Укр.

1.2. Програмні та технічні засоби організації онлайн сервісів

При розробці сучасного програмного забезпечення все більше уваги приділяється розробці такого типу додатків, як веб-сервіси. Інтернет вже давно домінує в сфері соціального обслуговування: за допомогою використання web-сервісів можна купувати квитки на потяги чи літак, оплачувати комунальні послуги чи отримувати адміністративні послуги, купувати побутову техніку. Це позбавляє населення від марної трати часу стоячи в чергах. У зв'язку зі вище сказаним, можна вважати, що розробка і застосування веб-сервісів для надання соціальних та адміністративних послуг, є досить актуальною темою в сучасному суспільстві.

Існує досить багато визначень поняття «веб-сервіс» [23-27]. Але найчастіше веб-сервіс визначається як веб-ресурс зі стандартизованими

інтерфейсами, який може взаємодіяти з іншими програмними модулями за допомогою повідомлень, заснованих на певних стандартах і протоколах» [25, с. 4]

Головною відмінністю веб-сервісу від веб-сайту є, то, що веб-сервіс може надавати певні послуги. Наприклад, такі веб-сервіси, як центри надання адміністративних послуг пропонує користувачеві певні послуги, які він може отримати безпосередньо на даному веб-сервісі. Веб-сайт - це сайт на якому розташовується будь-яка інформація, з якої користувач може ознайомитися.

Аналіз робіт присвячених розробці веб-сервісів дозволяє виділити два найбільш важливі напрямки: дизайн і навігація. Дизайн грає важливу роль в залученні уваги користувачів мережі Інтернет до веб-сервісу. Навігація в веб-сервісі повинна бути простою і зрозумілою - це позитивно позначається на конверсії і навіть на ранжируванні в пошукових системах.

Технології створення веб-сервісів, що базуються на можливостях Інтернету, покликані кардинально поліпшити взаємодію людей і інформаційних систем один з одним і забезпечити поєднання різних систем і процесів.

Технології створення веб-сервісів утворені з цілого ряду стандартних протоколів взаємодій, засобів опису моделей даних і інтерфейсів, а також допоміжних мережеслужб.

Виділяють кілька етапів розробки веб-сервісу: планування, реалізація, тестування, публікація, рекламування, супровід [28]: Розглянемо ці етапи детальніше:

- **Планування.** На цьому етапі повинні бути отримані відповіді на питання: мета створення веб-сервісу, аудиторія, на яку він розрахований, яка інформація буде розміщена. Доцільно познайомитися з сайтами, тематика яких близька до тематики

створюваного веб-сервісу, з тим, щоб уникнути помилок. Визначаються категорії, підрозділи сайту, кількість сторінок в кожному розділі, підрозділі. Розробляється структурна (навігаційна) схема сайту.

- Реалізація. На цьому етапі проводиться підготовка текстового та графічного матеріалу (друк, сканування, оптимізація). Матеріал розбивається по файлах відповідно до структури. Організуються посилання між веб-сторінками. Розробляється дизайн.
- Тестування. Після того як веб-сервіс створений, необхідно перевірити правильність переходів зі сторінки на сторінку.
- Публікація. Розміщення сайту в Internet. Для цього необхідно розмістити всі файли веб-сервісу на веб-сервері, що надає такі послуги.
- Рекламування. На цьому етапі передбачається взаємний обмін посиланнями з іншими сайтами, схожими за тематикою, реєстрація сайту в популярних пошукових серверах.
- Супровід. Після створення веб-сервісу потрібно не забувати оновлювати інформацію, розширювати матеріал, покращувати дизайн.

Сервісно-орієнтована архітектура (англ. SOA, service-oriented architecture) - являє собою модульний підхід до розробки програмного забезпечення, заснований на використанні сервісів зі стандартизованими інтерфейсами.

В основі SOA лежать принципи багаторазового використання функціональних елементів інформаційних технологій, уніфікації типових операційних процесів. Компоненти програми можуть бути розподілені по різних вузлах мережі і пропонуються як незалежні і слабо пов'язані сервіси-додатки.

Інтерфейс компонентів SOA-програми здійснює інкапсуляцію деталей реалізації конкретного компонента. SOA добре зарекомендувала себе при побудові великих корпоративних програмних систем. Цілий ряд розробників пропонують інструменти і рішення на основі SOA (наприклад, платформи Microsoft .NET, IBM Web Sphere, SAP Net Weaver, Diasoft і ін.).

Структура реалізації веб-сервісу, що містить у собі програмні елементи, їх зовнішні властивості і взаємозв'язки, називається архітектурою програми. Важливість програмної архітектури пояснюється тим, що в ній викладаються початкові проектні рішення, вона визначає обмеження реалізації та організаційну структуру, що розробляється, крім того, програмна архітектура дозволяє більш точно планувати ресурси, необхідні для створення програмного проекту.

Розглянемо докладно архітектуру веб-сервісів. Архітектура програмної системи охоплює не тільки її структурні і поведінкові аспекти, а й правила її використання та інтеграції з іншими системами, функціональність, продуктивність, гнучкість, надійність, можливість повторного застосування, повноту, економічні та технологічні обмеження.

У міру розвитку інформаційних технологій і програмних систем все більшого значення набуває їх інтеграція один з одним з метою побудови єдиного інформаційного простору. Як можна бачити з вищенаведених визначень інтеграція є найважливішим елементом архітектури. Для того щоб побудувати правильну і надійну архітектуру і грамотно спроектувати спільну роботу програмних систем необхідно чітко слідувати сучасним стандартам в області інформаційно-комунікаційних технологій.

Законодавцями стандартів в галузі інформаційно-комунікаційних технологій виступають такі міжнародні організації як SEI (Software Engineering Institute), WWW (консорціум World Wide Web), OMG (Object

Management Group), організація розробників Java - JCP (Java Community Process), IEEE (Institute of Electrical and Electronics Engineers) та інші.

Веб-сервіси це додатки з сервіс-орієнтованою архітектурою (Service-Oriented Architecture - SOA) [29, 30]. SOA являє собою модульний підхід до розробки програмного забезпечення, заснований на використанні сервісів (служб) зі стандартизованими інтерфейсами [31]. В даний час спостерігається стійке зростання інтересу до концепції сервіс-орієнтованої архітектури (Service-Oriented Architecture - SOA).

SOA [32] - це архітектурна модель для об'єднання доступних обчислювальних ресурсів, таких як: додатки, дані, форми їх подання, засоби управління і моніторингу, з вимогами досягнення бажаних результатів, для споживачів сервісів, якими можуть бути: кінцеві користувачі, інші додатки, або інші, подібні, сервіси. У загальному вигляді SOA [31, 33] передбачає наявність трьох основних учасників: постачальника сервісу, споживача сервісу та реєстру сервісів.

Взаємодія учасників виглядає досить просто: постачальник сервісу реєструє свої сервіси в реєстрі, а споживач звертається до сервісу із запитом. Організація з стандартизації OASIS [33] затвердила еталонну модель сервіс-орієнтованої архітектури Reference Model for Service-Oriented Architecture 1.0 (SOA-RM). Модель покликана ввести чітку технічну термінологію SOA для розробників. У. Мета моделі - надати загальну схему, яка зніме всі двозначності в різних реалізаціях SOA.

Робота веб-сервісів побудована на використанні декількох відкритих стандартів [34]: XML - розширювана мова розмітки, призначений для зберігання і передачі структурованих даних; SOAP - протокол обміну повідомленнями на базі XML; WSDL - мова опису зовнішніх інтерфейсів вебсервісів на базі XML; UDDI - універсальний інтерфейс розпізнавання, опису та інтеграції (Universal Discovery, Description, and Integration).

Зв'язок між веб-сервісами та їх клієнтами здійснюється за допомогою повідомлень в форматі XML. SOAP (Simple Object Access Protocol) являє собою протокол повідомлень для вибору веб-сервісів. SOAP [35-37] заснований на мові XML і розширює деякі протокол прикладного рівня: HTTP, FTP, SMTP і т.д.

Як правило, найчастіше використовується HTTP. Замість використання HTTP для запиту HTML-сторінки, яка буде показана в браузері, SOAP відправляє за допомогою HTTP-запиту XML-повідомлення і отримує результат у HTTP відповідь.

Для правильної обробки XML-повідомлення процес-«слухач» HTTP (напр. Apache або Microsoft IIS) повинен надати SOAP-процесор, або, іншими словами, повинен мати можливість обробляти XML.

SOAP [36] є найголовнішою частиною технології веб-сервісів. Він здійснює перенесення даних по мережі з одного місця в інше. SOAP забезпечує доставку даних веб-сервісів. Він дозволяє відправнику і одержувачу XML-документів підтримувати загальний протокол передачі даних, що забезпечує ефективність мережевого зв'язку. SOAP - це базова модель з'єднання, що забезпечує узгоджену передачу повідомлення від відправника до одержувача, потенційно допускає наявність посередників, які можуть обробляти частина повідомлення або додавати до нього додаткові елементи.

Специфікація SOAP містить угоди по перетворенню односпрямованого обміну повідомленнями відповідно до принципу «запит / відповідь», а також визначає, як здійснювати передачу всього XML-документа. Кожен веб-сервіс надає документ WSDL (Web Service Description Language - мова опису веб-сервісу), в якому описується все, що клієнту необхідно для роботи з цим сервісом.

WSDL-документ [38] надає простий і послідовний спосіб виклику будь-якого веб-методу. Більш того, цей документ дозволяє використовувати інструменти автоматичного генерування проксі-класів. WSDL документ заснований на XML форматі, відповідно до якого інформація поділяється на п'ять груп.

Специфікація UDDI [31] (Universal Description, Discovery, and Integration - універсальне опис, пошук і інтеграція) дозволяє уникнути різних проблем за допомогою використання спеціального сховища (сховищ), де підприємства та організації можуть розміщувати дані про їхні сервіси, включаючи Oracle і Microsoft.

Об'єднавши свої зусилля, ці компанії розробили проект специфікації UDDI, яка після закінчення 18 місяців була стандартизована. Інформація в цьому репозиторії повинна оновлюватися вручну. З цією метою деякі «вузлові оператори» зберігають ідентичні копії сховища UDDI. Ці компанії забезпечують зберігання зазначеного сховища та безкоштовний доступ до нього для популяризації веб-сервісів. Крім того, Майкрософт включила версію UDDI в програмне забезпечення сервера Windows .NET для використання в корпоративних мережах інтранету.

У сховищі UDDI містяться відомості про підприємства, що надають веб-сервіси, про тип кожного сервісу і зв'язках з інформацією і специфікаціями, що відносяться до цих сервісів. Інтерфейс UDDI сам по собі є веб-сервіс. Для реєстрації або пошуку служби слід відправити SOAP-повідомлення.

Головними недоліками веб-сервісів є менша продуктивність і більший розмір мережевого трафіку в порівнянні з такими технологіями як RMI, CORBA, DCOM за рахунок використання текстових XML-повідомлень.

Існує безліч платформ і мов програмування для створення веб-сервісів, розглянемо деякі з них. Мова програмування визначає набір лексичних,

синтаксичних і семантичних правил, які задають зовнішній вигляд програми і дії, які виконає виконавець (комп'ютер) під її управлінням [39]. Існує рейтинг найпопулярніших і затребуваних мов програмування. Найбільш популярних мов програмування на даний момент входять наступні мови: Java, C, C ++, Python, C #, Visual Basic .NET, Javascript, PHP, Perl [40-43].

Оскільки при реалізації прикладної програмної системи центру надання адміністративних була використана мова програмування PHP, розглянемо цей стандарт стандарт детальніше.

PHP (Personal Home Page) є широко використовуваною мовою програмування загального призначення з відкритим вихідним кодом, що впроваджується в HTML, інтенсивно застосовується для розробки веб-додатків. Його синтаксис заснований на синтаксисі мов C, Java і Perl і є легким для вивчення. PHP є одним з лідерів серед мов програмування. Головна мета мови полягає в наданні веб-розробникам можливості швидко створювати динамічно генеруються веб-сторінки [44].

Існують три основні області, де використовується PHP:

- Створення скриптів для виконання на стороні сервера, так званий серверний скриптинг. Це найбільш традиційна і головна сфера застосування PHP. PHP здатний виконувати все те, що виконує будь-яка інша програма CGI, наприклад, обробляти дані форм, генерувати динамічні сторінки. Все, що для цього знадобиться, це розбирач коду PHP (CGI або серверний модуль), веб-сервер і веб-браузер. Сервер повинен бути запущений і повинен мати з'єднання з інстальований PHP. Можна отримати висновок PHP-програми в веб-браузер, переглядаючи PHP-сторінку на сервері.
- Створення скриптів для виконання в командному рядку, так званий скриптинг командного рядка. Можна створити PHP-скрипт, здатний запускатися незалежно від веб-сервера та браузера. Все, що для цього

буде потрібно - розбирач PHP. Такий спосіб використання PHP ідеально підходить для скриптів, які повинні виконуватися регулярно. Ці скрипти також можуть бути використані в задачах простої обробки текстів.

- Створення клієнтських додатків GUI виконуються на стороні клієнта. Можливо, PHP є не найкращим мовою для створення подібних клієнтських додатків, але можна використовувати PHP-GTK, який є розширенням PHP для створення таких додатків. Подібним чином можна створювати і міжплатформні додатки.

PHP доступний для більшості операційних систем, включаючи Linux, багато модифікації Unix, Microsoft Windows, Mac OS X, RISC OS, та багатьох інших. Також в PHP включена підтримка більшості сучасних веб-серверів, таких, як Apache, Microsoft Internet Information Server, Personal Web Server, серверів Netscape і iPlanet, сервера Oreilly Website Pro, Caudium, Xitami, OmniHTTPd та багатьох інших [45-47].

Головні переваги PHP:

1. Практичність. PHP повинен надати програмістові засіб для швидкого і ефективного вирішення поставлених завдань. Практичний характер PHP обумовлений п'ятьма важливими характеристиками:

- традиційністю;
- простотою;
- ефективністю;
- безпекою;
- гнучкістю.

Існує ще одна «характеристика», яка робить PHP особливо привабливим: він розповсюджується безкоштовно. Причому, з відкритими вихідними кодами (Open Source).

2. Традиційність. Мова PHP знаком програмістам, що працюють в різних областях. Багато конструкції мови запозичені з Cі, Perl.

3. Простота.

4. Ефективність. Ефективність є виключно важливим чинником при програмуванні для розрахованих на багато користувачів середовищ, до числа яких належить і веб. Дуже важлива перевага PHP полягає в тому, що він дозволяє обробляти сценарії з достатньо високою швидкістю.

5. Безпека. PHP надає в розпорядження розробників та адміністраторів гнучкі та ефективні засоби безпеки, які умовно поділяються на дві категорії: засоби системного рівня та засоби рівня додатки.

- Засоби безпеки системного рівня. У PHP реалізовані механізми безпеки, що знаходяться під управлінням адміністраторів; при правильному підборі PHP це забезпечує максимальну свободу дій і безпеку. PHP може працювати в так званому безпечному режимі (safe mode), який обмежує можливості застосування PHP користувачами по ряду важливих показників. Наприклад, можна обмежити максимальний час виконання та використання пам'яті (неконтрольований витрата пам'яті негативно впливає на швидкодію сервера). Адміністратор також може встановлювати обмеження на каталоги, в яких користувач може переглядати і виконувати сценарії PHP, а також використовувати сценарії PHP для перегляду конфіденційної інформації на сервері [48,49].

- Засоби безпеки рівня додатка. У стандартний набір функцій PHP входить ряд надійних механізмів шифрування. PHP також сумісний з багатьма додатками незалежних фірм, що дозволяє легко інтегрувати його з захищеними технологіями електронної комерції. Інша перевага полягає в тому, що вихідний текст сценаріїв PHP можна переглянути у браузері, оскільки сценарій компілюється до його відправлення за запитом

користувача. Реалізація PHP на стороні сервера запобігає викрадення нетривіальних сценаріїв користувачами.

6. Гнучкість. Оскільки PHP є вбудованою мовою, він відрізняється винятковою гнучкістю по відношенню до потреб розробника. Хоча PHP зазвичай рекомендується використовувати у поєднанні з HTML, він з таким же успіхом інтегрується і в JavaScript, WML, XML та інші мови. Крім того, добре структуровані програми PHP легко розширюються в міру необхідності.

7. Безкоштовне розповсюдження .

Таким чином, вибираючи PHP, розробник отримує свободу вибору операційної системи і веб-сервера. Крім того, у нього з'являється вибір між використанням процедурного або об'єктно-орієнтованого програмування або ж їх поєднання [50-52].

PHP здатний не лише видавати HTML. Можливості PHP включають формування зображень, файлів PDF і навіть роликів Flash. PHP також здатний видавати будь-які текстові дані, такі, як XHTML та інші XML-файли. PHP здатний здійснювати автоматичну генерацію таких файлів і зберігати їх у файловій системі вашого сервера замість того, щоб віддавати клієнту, організовуючи, таким чином, кеш динамічний зміст, розташований на стороні сервера [53].

Одним із значних переваг PHP є підтримка широкого кола баз даних.

Табл. 1.2 відображає порівняння сучасних технологій розробки веб-сервісів.

Результати проведеного аналізу дозволяють зробити висновок про наявність переваг експлуатаційних характеристик PHP-технології . PHP фреймворки за останній час набрали популярність і стали базовою платформою для розробки веб- застосунків [53,54].

Таблиця 1.2

Порівняння сучасних технологій розробки

Технологія / Характеристика	PHP	JSP	JSP ASP.NET
Багатоплатформність	+	+	-
Продуктивність	+	+/-	+/-
Простота використання	+	+/-	+/-
Наявність доступних програмних бібліотек	+	+	+
Розподіл дизайну та логіки	+/-	+/-	+
Відкритий код	+	-	-

Використання цих систем дозволяє зекономити велику кількість часу, зменшити навантаження на процес розробки, позбавляючи від проблеми повторюваного коду, і швидко створювати якісні додатки. Застосування PHP фреймворків значно полегшує процес створення програми і робить його більш функціональним [55,56]. Порівняльний аналіз PHP фреймворків наведений в табл. 1.3

Таблиця 1.3

Порівняльний аналіз PHP фреймворків

Технологія Характеристика	Zend Framework 2	PHP Code	Igniter 2	Kohana 3	Symfony 3	Yii 2
Рівень проектів	Середні, великі	Маленькі, середні	Маленькі, великі	Маленькі, великі	Великі	Маленькі, великі
PHP 5.7, 7	+	-	+	+/-	+	+
Жорстка структура директорій	-	+	+	+	+	-
Встроєна підтримка інтернаціональності	+	+	+	+	+	+
Важкість при встановленні та налаштування	Висока	Низька	Низька	Низька	Висока	Середня
Потреба у додатковому налаштуванні	+	-	-	-	+	-
Повна підтримка ORM	-	+/-	-	+	+	+
Unit-тести для вихідного коду	+	+	-	-	+	+
Лицензія	New BSD	MIT	Своя	BSD-style	MIT	New BSD

1.3. Аналіз ефективності функціонування онлайн сервісів

Зазвичай ефективність функціонування онлайн сервісів оцінюється за позиціями, показниками Page Rank або тематичного індексу цитування сторінок ресурсу. Сьогодні ці метрики втрачають сенс, оскільки пошукові системи намагаються зробити результати пошуку індивідуальними для кожного користувача виходячи з його географічного регіону, минулих пошукових запитів, переходів на сайти з певною тематикою. Метрика Page Rank, яка ще 10 років тому показувала значимість сайту, на сьогодні є однією із сотень інших, які визначають якість сайту [57].

Метрики ефективності можна розділити на кілька категорій: економічні показники, пошукові показники, лояльність відвідувачів, аудиторія сайту та зміст сайту.

До економічних показників відносимо рівень конверсії, показник відмовлень, середню вартість та загальну кількість замовлень.

В пошукових показниках виділяємо наступні пункти: кількість відвідувачів з пошукових систем, позиції сайту за ключовими словами, значення Page Rank та кількість проіндексованих сторінок [58].

Категорія «лояльність відвідувачів» складається з середнього часу відвідування сайту та відвідування певної сторінки, кількості переглядів користувача, кількості повернень, кількості нових користувачів, кількості унікальних відвідувачів, кількості переглядів сайту.

Аудиторія сайту – це перелік країн-відвідувачів, джерело трафіку, мова користувача, технології.

До змісту сайту можна віднести швидкість завантаження сторінок, сторінки входу та виходу, зручність розташування контенту (з точки зору користувачів та спеціалістів).

Набір найважливіших показників є індивідуальним для різних веб-сервісів [59,60].

З точки зору аналізу інформаційних проектів та спеціалізованих сайтів, до найбільш інформативних показників ефективності можна віднести пошукові показники, показники лояльності користувачів, аудиторію сайту та його зміст [61].

Розглянемо ці показники більш детально [62-64]:

1. Пошукові показники інформують наскільки зручним є сайт з точки зору пошукових систем, оскільки це впливає на позиції сайту у результатах пошуку.

2. Лояльність відвідувачів показує наскільки зручним є сайт та зміст веб-сервісу відповідає очікуванням користувача, тобто якість контенту. Середній час відвідування веб-сервісу вказує на те, скільки часу користувач відвідував сторінки веб-сервісу.

3. Середня кількість переглядів сторінок на відвідувача свідчить про актуальність сайту для користувачів. Співвідношення один перегляд на одного відвідувача є показником того, що контент не зацікавив користувача.

4. Аналіз аудиторії веб-сервісу є одним з ключових показників ефективності. Ця інформація дозволяє змінювати та вдосконалювати веб-сервіс, орієнтуючись на користувача.

5. Зміст веб-сервісу дозволяє проаналізувати інформацію про те, на які сторінки потрапляє і що шукає користувач при відвідуванні сайту. Сторінки виходу необхідні для розуміння того, на якому етапі веб-сервіс стає неактуальним користувачу та він його покидає. Аналіз змісту є показником, який дуже важко оцінити, оскільки кожен користувач має власну думку з розташування певних блоків веб-сервісу.

Для того, щоб виявити ключові показники ефективності веб-сервісів, треба визначити цілі проекту [65]. Наприклад, для центрів надання

адміністративних послуг може бути зменшення кількості відмов, скорочення часу на оформлення тієї чи іншої послуги. При аналізі ефективності веб-сервісу слід одразу визначитися з показниками, які будуть вимірюватися. Для невеликого проекту достатньо до трьох показників, у багатофункціонального порталу їх може бути набагато більше [66-68].

З технічної точки зору важливими показниками ефективної роботи веб-сервісів є наступні:

- час відгуку;
- кількість користувачів, які одночасно працюють із системою;
- запити за секунду;
- транзакції за секунду.

Час відгуку – це період, за який сервер відповідає на користувацький запит (швидкість завантаження сторінки, видачі пошукових запитів, підрахунків та аналіз даних) [69-72].

При тривалому очікуванні відповіді користувач втрачає час. Це знижує зацікавленість взаємодії з продуктом. Дану інформацію можна знайти на дослідницькому порталі Think with Google. Йдеться про те, що у випадку, коли час завантаження сторінки перевищує 3 секунди, то близько 53% відвідувачів готові закрити сайт і перейти на інший. Тривале завантаження також впливає на конверсію, знижуючи її майже на 20% за кожні 3-5 секунд користувацького очікування. Збільшення часу відгуку залежить від числа кількості користувачів, які одночасно працюють із системою, складності запиту. Щоб отримати об'єктивну інформацію про програмний продукт, варто проаналізувати і середньостатистичні показники. Вони формуються на основі аналізу безлічі циклів відправки запитів і отримання відповідей. При визначенні середньостатистичного часу відгуку враховуються швидкість доставки файлів (відображення тексту, картинок чи відеоматеріалів) і віддаленість користувача від цільового веб-сервера [73-75].

Але навіть при задовільних середніх показниках може статися відтік відвідувачів. У цьому випадку варто звернутися до аналізу пікових показників часу завантаження. Пікові значення – це найбільш високі, що виходять за рамки середніх, значення часу, при яких користувач проводить в очікуванні. Даний показник дозволяє визначити проблемне місце в роботі системи. Для оперативного виявлення проблем рекомендується оцінювати пікові та середні швидкості завантаження [76].

Кількість користувачів, які одночасно працюють із системою. Досягти максимальної реалістичності навантаження і отримати ефект, відповідний фактичній продуктивності системи, можна за допомогою віртуальних користувачів. Визначення максимального числа паралельно до працюючих користувачів актуально не тільки напередодні очікуваного збільшення потоку клієнтів, а й для кращого розуміння можливостей ПЗ. Якщо вузьке місце веб-сервісу, нездатність функціонувати при великому числі користувачів виявляються вже в момент напливу користувачів, то зростає ризик часткового або повного припинення роботи програмного продукту. Запобігти подібним проблемам можна в рамках тестування за допомогою імітації критичного навантаження, що виявляє граничні показники стійкості системи. Своєчасне тестування допоможе заощадити кошти і уникнути репутаційних провалів [77,78].

Запити за секунду – це число запитів, яке успішно обробляється за 1 секунду. У цьому випадку враховуються всі можливі варіанти взаємодії користувача з веб-сервісом: пошук інформації, завантаження або вивантаження даних. Максимальне число оброблених в секунду запитів залежить від кількості ресурсів на сторінці і їх типу. Наприклад, запит на відображення невеликого за розміром зображення буде оброблятися швидше, ніж на великий текстовий блок [79].

Чим швидше виконується кожен запит окремо, тим більше запитів можливо обробити за секунду. Для користувача цей показник конвертується в швидкість взаємодії з елементами веб-сервісу. Його потрібно враховувати при оцінці якості роботи ПЗ. Адже низькі показники числа виконаних за секунду запитів можуть обернутися незадовільною роботою сервера при зростаючому інтересі до веб-сервісу.

Транзакції за секунду – показник відображає число успішно завершених за одиницю часу транзакцій, операцій, які згруповані в логічній послідовності.

Своєчасне тестування програмного продукту допомагає виявляти проблемні ділянки роботи системи, які негативно позначаються на клієнтському досвіді (затримки при роботі, некоректне виконання запитів, помилки та інше), а також передбачати появу проблем [80].

1.4. Постановка задачі дисертаційного дослідження

Одним із визначальних чинників процесу створення ефективних ЦНАП та підвищення якості надання адміністративних послуг є використання сучасних інформаційних технологій. Ефективна інформаційна система надання адміністративних послуг дасть змогу одержувати об'єктивну інформацію щодо якості надання послуг. Використання інтелектуальних методів обробки даних може стати основою для оперативного ухвалення управлінських рішень, покращити якість послуг, наблизити їх до потреб замовників та значно прискорити надання адміністративних послуг для громадян.

Таким чином, актуальною є науково-практична задача розробки математичного та програмного забезпечення підтримки веб-сервісів надання адміністративних послуг шляхом впровадження засобів інтелектуалізації

процесу виконання послуг та математичного моделювання ефективності функціонування веб-сервісів.

Тому основними задачами, що виникають в процесі розробки інтелектуального модуля надання адміністративних послуг та математичного моделювання ефективності функціонування веб-сервісів, є наступні:

1. Запропонувати та обґрунтувати показник ефективності функціонування веб-сервісів у вигляді динаміки відвідуваності користувачами та його математичну модель, яка, на відміну від існуючих, побудована на методах аналізу інтервальних даних та врахування циклічності процесів відвідуваності.

2. Розробити метод інтелектуального опрацювання інформації для оцінювання подібності онлайн послуг, який ґрунтується на кластеризації документів.

3. Удосконалити метод автоматизованого пошуку оптимальних наборів документів в заданій предметній області з інтелектуальним модулем.

4. Розробити архітектуру прикладної програмної системи центрів надання адміністративних послуг для рівня муніципалітетів, яка містить інтелектуальні модулі обробки інформації про надавані послуги та модулі оптимізації процесів виконання цих послуг за часом виконання

5. Провести експериментальні дослідження та здійснити оцінку ефективності запропонованих методів та засобів.

Висновки до розділу 1

1. Проведено аналіз відомих інформаційних систем щодо надання адміністративних послуг. Виділено структуру основних показників, які використовуються для оцінки ефективності інформаційних систем даної

предметної області. Досліджено основні проблеми, які виникають та сформовано основний напрямок їх розв'язку.

2. Проаналізовано програмні засоби реалізації веб-орієнтованих додатків, включаючи особливості їх використання для систем заданого типу. Визначено переваги стеку технологій PHP, MySQL, фреймворк Zend над іншими засобами даного типу.

3. Здійснено оцінку ефективності функціонування онлайн сервісів. Виділено основні метрики ефективності, проаналізовано основні інструменти оцінки показників зазначених метрик. Показано вплив показників функціонування веб-ресурсів на ефективність функціонування інформаційної системи ЦНАП.

4 Зважаючи на особливості функціонування інформаційних систем ЦНАП, які викликані специфічною структурою бізнес-процесів, та недоліки існуючих методів, засобів інтелектуального аналізу даних і відповідних інтелектуальних модулів, актуальною є розробка математичного й програмного забезпечення підтримки веб-сервісів надання адміністративних послуг шляхом впровадження засобів інтелектуалізації процесу виконання послуг та математичного моделювання ефективності функціонування веб-сервісів.

РОЗДІЛ 2

ІНТЕРВАЛЬНА МОДЕЛЬ ЕФЕКТИВНОСТІ ФУНКЦІОНУВАННЯ ІНФОРМАЦІЙНИХ ВЕБ-РЕСУРСІВ

Цифровізація усіх сфер людської діяльності є пріоритетним розвитком людства у найближче десятиліття. Сьогодні різноманітні веб-ресурси створюють можливості для користувачів і віртуалізують процеси створення та надання різноманітних послуг. Проте, функціональність та ефективність багатьох онлайн сервісів вимагає суттєвого покращення. Однією із основних характеристик сервісу і загалом інформаційного веб-ресурсу є відвідуваність користувачами. Чим вища відвідуваність, тим більш актуальним є ресурс. Разом з тим показник відвідуваності он-лайн сервісу може свідчити про навантаження на персонал, який надає послуги, відображати потреби щодо оптимізації бізнес процесів в організації.

У даному розділі обґрунтовано показник ефективності веб-ресурсів у вигляді активності користувачів (кількості відвідувань). Для його дослідження, оптимізації та прогнозування запропоновано використовувати інтервальні моделі у вигляді різницевого рівняння. Обґрунтовано та запропоновано для побудови зазначених моделей використати експериментальні дані у вигляді числових інтервалів, методи аналізу інтервальних даних, а для ідентифікації параметрів моделі – методи параметричної ідентифікації на основі алгоритмів штучної бджолоїної колонії.

У підрозділах запропоновано верифікацію розроблених методів на прикладі побудови інтервальної моделі ефективності веб-ресурсу для надання адміністративних послуг ЦНАП Тернопільської міської ради. В завершальній частині розділу сформульовано основні наукові та практичні результати, наведених у цьому розділі досліджень.

Таким чином, у даному розділі розглянуто математичні моделі динаміки ефективності функціонування інформаційних веб-ресурсів. Проведено ряд чисельних експериментальних досліджень, які виконані на основі даних функціонування конкретного веб-ресурсу. В результаті проведених експериментів підтверджено ефективність застосування запропонованих моделей.

Матеріали розділу ґрунтуються на опублікованих автором працях [1, 2, 3, 6, 7, 9, 12].

2.1. Обґрунтування показника ефективності функціонування інформаційних веб-ресурсів

Особливим попитом користуються віртуальні системи надання адміністративних послуг, зокрема, пов'язані із веденням бізнесу, започаткуванням виробництва товарів та послуг. У першу чергу це стосується послуг щодо екологічної експертизи і надання дозвільної документації для розгортання промислового виробництва. Такі системи функціонують у розпорядженні муніципалітетів і суттєво спрощують процеси надання адміністративних послуг. Розробниками однієї з таких систем (www.snar.rada.te.ua) є автори даної праці. Важливим у цьому є ефективність функціонування інформаційних веб-ресурсів. Часто створювані системи є низько ефективними через слабе структурування контенту, не врахування динаміки запитів та активності користувачів.

Процеси взаємодії з інформаційними джерелами полягають у формуванні запитів, перетворенні їх в термінах джерел, отриманих та опрацьованих даних від джерел і формування результату, який за допомогою веб-сервера буде відправлений веб-браузеру. Це здійснюється підсистемою доступу до даних веб-ресурсу. Для взаємодії з джерелами використовується

схема даних, яка містить опис всіх підключених до ресурсу джерел, опис типів даних, що реалізуються джерелами, і правила формування запитів до джерел в термінах цих джерел. Разом з тим віртуальні системи надання адміністративних послуг докорінно відрізняються від звичайних інформаційних веб ресурсів, на яких добре структурований контент за типами послуг та документами, які необхідно завантажувати користувачеві. Тут важливою є аналітика отримана із веб ресурсу, яка необхідна для організації бізнес процесів персоналом центру надання адміністративних послуг. Такою аналітикою може бути загальна активність користувачів або їх активність у розрізі видів послуг.

Для оцінки ефективності функціонування інформаційних веб-ресурсів потрібні математичні моделі процесів їх функціонування. Вони дозволяють вирішувати як задачі оптимізації програмних алгоритмів і використання технічних ресурсів, так і задачі оптимізації організації бізнес процесів персоналом. Побудова таких моделей вимагає всебічного дослідження процесів функціонування веб-ресурсів з урахуванням впливу всіх елементів системи на її функціонування [81-83].

Для моделі ефективності інформаційних джерел в якості вихідного показника достатньо часто обирають величину часу, витрачену джерелом на виконання запитів на поточному інтервалі спостереження [84-87]. Для моделі активності користувачів зазвичай вихідним показником обирають число запитів до інформаційних джерел на поточному інтервалі спостереження, при цьому зовнішні впливи не спостерігаються. У контексті вищезазначеного, модель активності користувачів є переважаючою для цього типу веб ресурсів.

Важливою вимогою до математичних моделей є її відповідність (адекватність) до досліджуваного інформаційного веб-ресурсу. Тому її доцільно будувати на основі індуктивного підходу, коли створений веб-

ресурс функціонує у тестовому чи експериментальному режимі, а отримані дані є основою для побудови моделі його ефективності [88-89]. Слід брати до уваги той факт, що в процесі функціонування веб-ресурсу на його показники впливає ряд неконтрольованих чинників, які спотворюють результати спостережень. Для часових характеристик відгуку на запити впливають такі чинники: різна продуктивність апаратного і програмного забезпечення, використовуваного джерелами; різний обсяг даних джерела; різниця в обсязі результатів виконання запитів, пов'язана, в тому числі з складністю конкретного запиту; різна доступність джерел, пов'язана з характером роботи середовища; різний рівень навантаження на апаратне та системне програмне забезпечення з боку інших додатків, які існують на тій же платформі; різна завантаженість сегментів мережі для використання в взаємодії веб-ресурсу з джерелом тощо. Частина тих чинників також має вплив на активність користувачів, наприклад, сезонність та періодичність. Саме цей факт беремо за основу побудови моделі активності користувачів. Вплив неконтрольованих чинників на інтервал сезонних коливань пропонується враховувати у вигляді інтервальних оцінок характеристики, яка спостерігається. Такий підхід дозволить відслідковувати та прогнозувати динаміку ефективності веб-ресурсу з гарантованою точністю в межах дії неврахованих та неконтрольованих чинників.

Як зазначено вище, ефективність веб-ресурсу будемо описувати активністю користувачів, яка має чисто виражений циклічний характер.

При побудові моделі описаного явища повинні бути враховані наступні вимоги:

- сформована модель повинна враховувати залежність модельованого показника від часу (бути динамічною), оскільки, як було зазначено вище, активність користувачів змінюється в часі і має періодичний характер, що

важливо з точки зору аналізу та оптимізації бізнес процесів персоналу центру надання адмін послуг;

- неконтрольовані чинники, які спотворюють результати спостережень за часовими характеристиками активності користувачів, пропонується враховувати у вигляді інтервальних оцінок характеристики, яка спостерігається. Це дозволить відслідковувати та прогнозувати динаміку ефективності веб-ресурсу з гарантованою точністю в межах дії неврахованих та неконтрольованих чинників;

- при моделюванні має використовуватися дискретний час – інтервали спостереження повинні бути сформовані таким чином, щоб обчислення необхідної оцінки можна було робити в межах інтервалу, а використовувані моделлю обчислювальні ресурси були досить малі в порівнянні з тими, які необхідні для забезпечення основної функціональності веб-ресурсу.

За рахунок зміни параметрів відповідних моделей стає можливим моделювати різні види інформаційних джерел і режими активності користувача. Як результат, з'являється можливість аналізувати працездатність і ефективність веб-системи з різними типами інформаційних джерел і режимами активності користувача при різних комбінаціях налаштувань.

2.2. Інтервальна модель ефективності функціонування інформаційних веб-ресурсів та метод її ідентифікації

За рахунок зміни параметрів відповідних моделей стає можливим моделювати різні види інформаційних джерел і режими активності користувача. Як результат, з'являється можливість аналізувати працездатність і ефективність веб-системи з різними режимами активності користувача, при різних комбінаціях налаштувань.

Для кожної категорії послуг модель спостережень можна представити у такий спосіб:

$$[z_{lk}^-; z_{lk}^+], k = 0, \dots, K, \quad (2.1)$$

Де $[z_{lk}^-; z_{lk}^+]$ відповідно, нижня та верхня межі інтервалу можливих значень активності користувачів до певного виду послуг інформаційного ресурсу на часовій дискреті $k = 0, \dots, K$.

Для лінійної за параметрами дискретної моделі активності користувачів до певного l - того виду послуг інформаційного ресурсу використаємо такий вираз:

$$v_k = \vec{f}^T(v_{k-d}, \dots, v_{k-1}, \vec{u}_0, \dots, \vec{u}_k) \cdot \vec{g}, k = d, \dots, K, \quad (2.2)$$

Де $\vec{f}^T$ вектор базисних функцій, у загальному випадку нелінійних, з допомогою яких виконують перетворення значень модельованої характеристики (значень активності користувачів), а також вхідних змінних для певних часових дискрет; набори векторів вхідних змінних (управлінь) $\vec{u}_0, \dots, \vec{u}_k$; d - порядок дискретної моделі, який як відомо є еквівалентним порядку диференціального рівняння – аналогу дискретної моделі.

Спираючись на вимоги забезпечення точності математичної моделі в межах точності спостережень, умови узгодження експериментальних даних, представлених в інтервальному вигляді (2.1), із даними отриманими на основі математичної моделі у вигляді дискретної моделі (2.2) сформулюємо у такому вигляді:

$$v_k = \vec{f}^T(v_{k-d}, \dots, v_{k-1}, \vec{u}_0, \dots, \vec{u}_k) \cdot \vec{g}, k = d, \dots, K. \quad (2.3)$$

Умови (2.3) забезпечують отримання обчислених значень активності користувачів до інформаційних веб-ресурсів в межах інтервалів можливих значень цієї кількості активних відвідувачів веб-ресурсу, які ми отримуємо при дослідженні сайту.

Спираючись на результати проведеного аналізу можемо констатувати, що для забезпечення умов (2.3) заданої точності макромоделі у вигляді лінійних різницевих операторів (2.2) при розв'язуванні задачі її параметричної ідентифікації обґрунтованим є застосування методів аналізу інтервальних даних [87,88].

Припустимо, що вектор оцінок $\hat{\vec{g}}$ параметрів $\vec{g}$ у різницевому операторі (2) отримано на основі аналізу інтервальних даних. Підставляючи вектор оцінок параметрів $\hat{\vec{g}}$ різницевого оператора замість вектора їх істинних значень $\vec{g}$ у вираз (2.2) разом із заданими початковими інтервальними значеннями кожного елементу із набору $[\hat{v}_0], \dots, [\hat{v}_{k-d}]$ та заданими векторами вхідних змінних $\vec{u}_0, \dots, \vec{u}_k$ отримаємо інтервальну оцінку модельованої характеристики $[\hat{v}_k]$ на часових дискретах : $k = d, \dots, K$

$$[\hat{v}_k] = [\hat{v}_k^-; \hat{v}_k^+] = \vec{f}^T([\hat{v}_{k-d}], \dots, [\hat{v}_{k-1}], \vec{u}_0, \dots, \vec{u}_k) \cdot \hat{\vec{g}}, k = d, \dots, K. \quad (2.4)$$

Таким чином кількість активних користувачів будемо прогнозувати на основі інтервальної дискретної моделі (ІДМ) [87].

Для математичного формулювання задачі параметричної ідентифікації ІДМ (4) на основі аналізу інтервальних даних запишемо умови узгодження даних спостережень, представлених в інтервальному вигляді (2.1), із даними отриманими на основі ІДМ запишемо у такому вигляді:

$$[\hat{v}_k^-; \hat{v}_k^+] \subset [z_k^-; z_k^+], \forall k = 0, \dots, K. \quad (2.5)$$

Умови (2.5) забезпечують отримання інтервальних оцінок кількості активних користувачів до інформаційних веб-ресурсів в межах інтервалів, отриманих з результатів спостережень за активністю користувачів веб-ресурсу.

Підставимо у вирази (2.5) замість інтервальних оцінок $[\hat{v}_{ik}^-; \hat{v}_{ik}^+]$ модельованої характеристики її інтервальні значення, обчислені на основі

ІДМ (4) разом із урахуванням заданих початкових інтервальних значень кожного елементу із набору:

$$[\hat{v}_{l0}] \subseteq [z_{l0}], \dots, [\hat{v}_{lk-d}] \subseteq [z_{lk-d}], \quad (2.6)$$

та заданими векторами вхідних змінних $\vec{u}_0, \dots, \vec{u}_k$, отримаємо:

$$\begin{cases} [\hat{v}_{l0}^-; \hat{v}_{l0}^+] \subseteq [z_{l0}^-; z_{l0}^+], \dots, [\hat{v}_{lk-d}^-; \hat{v}_{lk-d}^+] \subseteq [z_{lk-d}^-; z_{lk-d}^+]; \\ z_{lk}^- \leq \vec{f}^T([\hat{v}_{lk-d}], \dots, [\hat{v}_{lk-1}], \vec{u}_0, \dots, \vec{u}_k) \cdot \hat{g}_l \leq z_{lk}^+; \\ k = d, \dots, K. \end{cases} \quad (2.7)$$

Отже, ІСНАР (7) для кожного l – того інформаційного ресурсу отримуємо, підставляючи інтервальні оцінки вихідної характеристики $[\hat{v}_{lk}^-; \hat{v}_{lk}^+]$ (задані у вигляді початкових умов та прогнозовані на основі виразу (2.6) у решті вузлів сітки) в умови (2.5).

Як відомо, отримані системи є інтервальними системами нелінійних алгебричних рівнянь (ІСНАР). Отже, задача ідентифікації параметрів ІДМ (2.4) за умов (2.5) є задачею розв'язування ІСНАР у вигляді (2.7).

Враховуючи, що зазначена задача не може бути розв'язана за наперед задану кількість ітерацій, такого типу задачі відносять до NP-повних. Єдиним підходом для їх розв'язування є методи повного перебору або методи випадкового пошуку. Зважаючи на складність задачі параметричної ідентифікації макромоделі у вигляді ІДМ, для пошуку хоча б одного розв'язку ІСНАР можемо використати методи випадкового пошуку [83].

Для моделі ефективності інформаційних джерел обрано активності користувачів з вихідним показником числа запитів до інформаційних джерел на поточному інтервалі спостереження, при цьому зовнішні впливи не спостерігаються.

Оскільки для досліджуваного показника складно запропонувати моделі, побудовані на основі фізичних аналогій, їх передбачається ідентифікувати. При цьому для лінійної за параметрами дискретної моделі (2.2) необхідно розв'язувати задачу структурної та параметричної

ідентифікації [83,87]. Нелінійні процеси динаміки активності користувачів до певного виду інформаційних ресурсів можна врахувати різницеvim оператором у загальному вигляді (2.2). У нашому випадку структуру моделі будемо задавати, а розв'язуватимемо тільки задачу параметричної ідентифікації.

Таким чином, метою досліджень є параметрична ідентифікація ІДМ активності користувачів до певного виду інформаційних ресурсів і на основі цієї моделі в подальшому проведення оптимізації структури цих ресурсів.

Отримати повний розв'язок ІСНАР (2.7) у вигляді множини значень компонент вектора $\vec{g}$, в загальному випадку є не можливо. Тому за допомогою ітераційної процедури будемо шукати хоча б один розв'язок, який забезпечує належність до прогнозованого інтервалу кількості активних користувачів веб ресурсу для заданих дискрет.

Ітераційна процедура розв'язуватиме оптимізаційну задачу, яку сформуємо із ІСЛАР (2.7). Показник «якості» поточного розв'язку ІСНАР відображає ступінь наближеності цього розв'язку, до розв'язку, який перетворює ІСНАР (2.7) у сумісну систему. Очевидно, що «якість» наближення буде тим вищою, чим ближче буде прогнозований коридор, побудований на основі даного наближення вектора параметрів, до експериментального.

Якість наближення будемо визначати у вигляді різниці центрів найбільш віддалених між собою прогнозного та експериментального інтервалів – у випадку, коли вони не перетинаються та найменшою шириною перетину серед прогнозних та експериментальних інтервалів – для випадку їх перетину [87,89]:

$$\delta(\hat{g}_l) = \max_{k=d, \dots, K} \{ |mid([\hat{v}_k]) - mid([z_k])| \}, \text{ якщо } [\hat{v}_k] \cap [z_k] = \emptyset, \exists k = d, \dots, K; \quad (2.8)$$

$$\delta(\hat{g}_l) = \max_{k=d, \dots, K} \{ wid([\hat{v}_k]) - wid([\hat{v}_k] \cap [z_k]) \}, \text{ якщо } [\hat{v}_k] \cap [z_k] \neq \emptyset, \forall k = d, \dots, K, \quad (2.9)$$

де $mid(\bullet)$ та $wid(\bullet)$, - операції визначення центру та ширини інтервалу, відповідно.

$$\delta(\hat{g}_l) \xrightarrow{\hat{g}_l} \min, \hat{g}_{il} \in [g_{li}^{low}, g_{li}^{up}], i = 1, \dots, n, l = 1, \dots, S, \quad (2.10)$$

де g_{li}^{low} , g_{li}^{up} - найменше та найбільше значення кожного параметра ІДМ.

Таким чином, для функції мети, яка визначає «якість» поточного наближення матимемо два вирази (2.8) та (2.9), які обчислюють «грубу» та точну оцінки вектора параметрів, відповідно.

Функції (2.8) та (2.9) є достатньо складними, дискретними. Обчислення навіть одного їх значення є достатньо трудомісткою процедурою, з NP складністю.

Якщо обчислене значення «якості» $\delta(\hat{g}_l)$ поточного наближення оцінки $\hat{g}_l$ вектора параметрів ІДМ на поточній ітерації дорівнює нулю ($\delta(\hat{g}_l) = 0$), то – завершення процедури (10).

Зазначену задачу не можливо розв'язати за наперед задану кількість ітерацій. Єдиним підходом для їх розв'язування є методи повного перебору, або евристичні методи, зокрема методи випадкового пошуку [83].

В основі методів випадкового пошуку у контексті розв'язування задачі параметричної ідентифікації ІДМ покладено ітераційні процедури [83]: задання початкових умов у вигляді (6); задання початкової $\hat{g}_0$ чи формування випадковим чином поточної $\hat{g}_l$ оцінки вектора параметрів різницевого оператора; обчислення інтервальних оцінок модельованої характеристики $[\hat{v}_k]$ у точках на часових дискретах $k = d, \dots, K$ із застосуванням рекурентної схеми; перевірка «якості» $\delta(\hat{g}_l)$ поточного наближення оцінки $\hat{g}$ вектора параметрів ІДМ. При формуванні поточної оцінки вектора параметрів різницевого оператора використовують різні

підходи, серед яких і випадковий пошук чи генетичні оператори (схрещування), або оператори систем із самоорганізацією. Разом з тим, існуючі методи не придатні для розв'язування задач із такими складними функціями мети, які задано виразами (8) чи (9). За цих умов пропонується використати алгоритми бджолиної колонії (АБК) [90,91]. Вони є найбільш придатні для розв'язування оптимізаційних задач зі складними нелінійними та дискретними функціями мети. АБК складаються із чотирьох фаз: ініціалізації; робочих бджіл; бджіл дослідників; бджіл розвідників. Розглянемо цей алгоритм детально [89,90–92].

Фаза ініціалізації. Вектори, які визначають координати джерел нектару для однієї l -тої бджоли, позначаємо за $\hat{g}_l$. В контексті задачі оцінювання параметрів це i є вектори параметрів ІДМ. Чисельність усієї популяції приймаємо за S . Всі вектори популяції ініціалізуються ($l=1, \dots, S$) бджолами - розвідниками. Кожен вектор параметрів містить n змінних $g_{li}, i = 1, \dots, n$, які слід оцінити за критерієм (2.10).

На цій фазі для ініціалізації випадковим чином початкових джерел нектару використовуємо таку формулу [92,93]:

$$g_{li} = g_{li}^{low} + rand(0,1) * (g_{li}^{up} - g_{li}^{low}), i = 1, \dots, n, l = 1, \dots, S, \quad (2.11)$$

де $rand(0,1)$ випадкове дійсне число із діапазону $(0,1)$.

Варто зазначити, що на цій фазі також налаштовуємо усі параметри алгоритму, такі як $mcp=0$ – поточний номер ітерації та MCN – загальна кількість ітерацій, $LIMIT$ – число, яке визначає вичерпність джерела та його поточне значення $limit=0$.

Фаза робочих бджіл. Робочі бджоли здійснюють пошук нових джерел з більшою кількістю нектару. В контексті оптимізаційної задачі (2.10) це означає пошук нових розв'язків з меншими значеннями функції мети (2,8) чи (2.9). Отже, робочі бджоли шукають сусідні по відношенню до поточних

джерел нектару та оцінюють їх «якість». Для обчислення сусідніх джерел нектару використовуємо такі формули [92,94]:

$$g_{li}^{mcn} = g_{li} + \Phi_{li} * (g_{li} - g_{pi}), i = 1, \dots, n, p \neq l = 1, \dots, S, \quad (2.12)$$

де g_{li}^{mcn} - координата джерела нектару на поточній ітерації $\bar{g}_p$ - випадково обраний вектор джерела нектару із $p \neq l = 1, \dots, S$; $i = 1, \dots, n$ - випадково обраний індекс параметра; Φ_{li} - випадкове число із діапазону $[-1, 1]$.

Після обчислення координат нових джерел нектару $\bar{g}_l^{mcn}$ проводиться попарна селекція між існуючими та поточними із застосуванням функції мети (8) чи (9):

$$\hat{g}_l = \{\hat{g}_l, \text{якщо } \delta(\hat{g}_l) \leq \delta(\bar{g}_l^{mcn})\} \text{ or } \hat{g}_l = \{\bar{g}_l^{mcn}, \text{якщо } \delta(\hat{g}_l) > \delta(\bar{g}_l^{mcn})\}. \quad (2.13)$$

Фаза бджіл дослідників. Згідно з АБК [90,91], неробочі бджоли складаються з двох груп: дослідників та розвідників. Робочі бджоли прилітають до вулика і діляться у вулику своєю інформацією про джерела нектару з бджолами дослідниками. Останні роблять вибір джерел нектару за ймовірністю, залежно від отриманої інформації. Отже, для кожного поточного джерела нектару розраховуємо ймовірність P_l його вибору бджолами дослідниками [92,93]:

$$P_l = \frac{1 - \delta(\hat{g}_l)}{\sum_{l=1}^S (1 - \delta(\hat{g}_l))}. \quad (2.14)$$

При цьому, перед використанням формули (2.14) попередньо нормуємо значення функції мети $\delta(\hat{g}_l)$ до одиничного інтервалу $[0, 1]$.

На підставі розрахованих ймовірностей, бджоли розвідники обирають окіл з m_l –точок певних джерел нектару $\hat{g}_l$, координати яких обчислені за формулою (2.12). При цьому, кількість точок околу для кожного джерела обчислюємо за формулою [14,15]:

$$m_l = P_l \cdot S. \quad (2.15)$$

Для кожного згенерованого джерела нектару обчислюємо функцію мети (8) чи (9) і знову проводимо попарну селекцію між існуючими та поточними джерелами нектару на основі виразу (2.13).

Фаза бджіл розвідників. Це фаза бджіл, які обирають нові джерела нектару випадковим чином [94-97]. На цій фазі параметри ІДМ обчислюємо із застосуванням виразу (2.11). Дана процедура відбувається наступним чином: робочі бджоли, які формують нові розв'язки задачі оптимізації, стають бджолами розвідниками, якщо ці розв'язки не покращуються протягом багатьох ітерацій. В контексті поведінкової моделі бджолиної колонії це означає вичерпання поточних джерел нектару, коли лічильник *limit* перевищує його межеве значення *LIMIT*.

З точки зору розв'язування оптимізаційної задачі (10) механізм із застосуванням бджіл розвідників забезпечує вихід із локальних мінімумів [98-101].

Реалізація методу параметричної ідентифікації виконана у системі, яка складається з трьох підсистем, що взаємодіють через RESTful API, зокрема:

1) Підсистема реєстрації у вигляді веб-ресурсу, яка дозволяє додавати нових користувачів і додавати нові математичні вирази (структури), які потрібно обрахувати. Є можливість задавати налаштування процесу обрахунку, бачити проміжні та кінцеві результати обчислень. Дана підсистема розроблена на мові програмування PHP з використанням Laravel фреймворка, СУБД MySQL. На клієнтській частині сайту (frontend) використовується Bootstrap CSS Framework, JavaScript-бібліотека jQuery.

2) Підсистема, в якій виконуються математичні обрахунки, розроблена з використанням мови програмування PHP, Lumen мікро-фреймворка, СУБД MySQL. Використані програмні бібліотеки для роботи з математичними виразами (MathExecutor) та інтервалами (php-math-interval).

3) Підсистема збору, пошуку та візуалізації даних, де можна переглянути як саме ведуться обчислення, які точки обираються на кожній ітерації циклу та іншу інформацію, що може допомогти контролювати процес обчислення. Це дає можливість більш ефективно виявляти помилки та знаходити оптимальніші способи обрахунку. В якості бази даних використовується Elasticsearch, а для опрацювання даних – Kiban.

2.3. Методика отримання експериментальних даних

Таким чином, для побудови інтервальної математичної моделі ефективності функціонування інформаційних веб-ресурсів, в основі якої є показник відвідуваності користувачами, необхідно розв'язати задачу ідентифікації її параметрів. Процедура ідентифікації параметрів вимагає отримання результатів експерименту для конкретного інформаційного веб-ресурсу, до якого дослідник має не тільки доступ, але і вихідний програмний код.

Для отримання кількості переглядів визначеної сторінки на сайті використовується Google API (Analytics Reporting 4). Процедура отримання інформації через Google API представлено на рисунку 2.1.

У цьому випадку нам необхідно в запиті встановити:

- діапазон дат;
- показник: `ga:pageviews` (кількість переглядів);
- фільтр по параметру `ga:pagePath`, значення якого має дорівнювати URL певної сторінки.

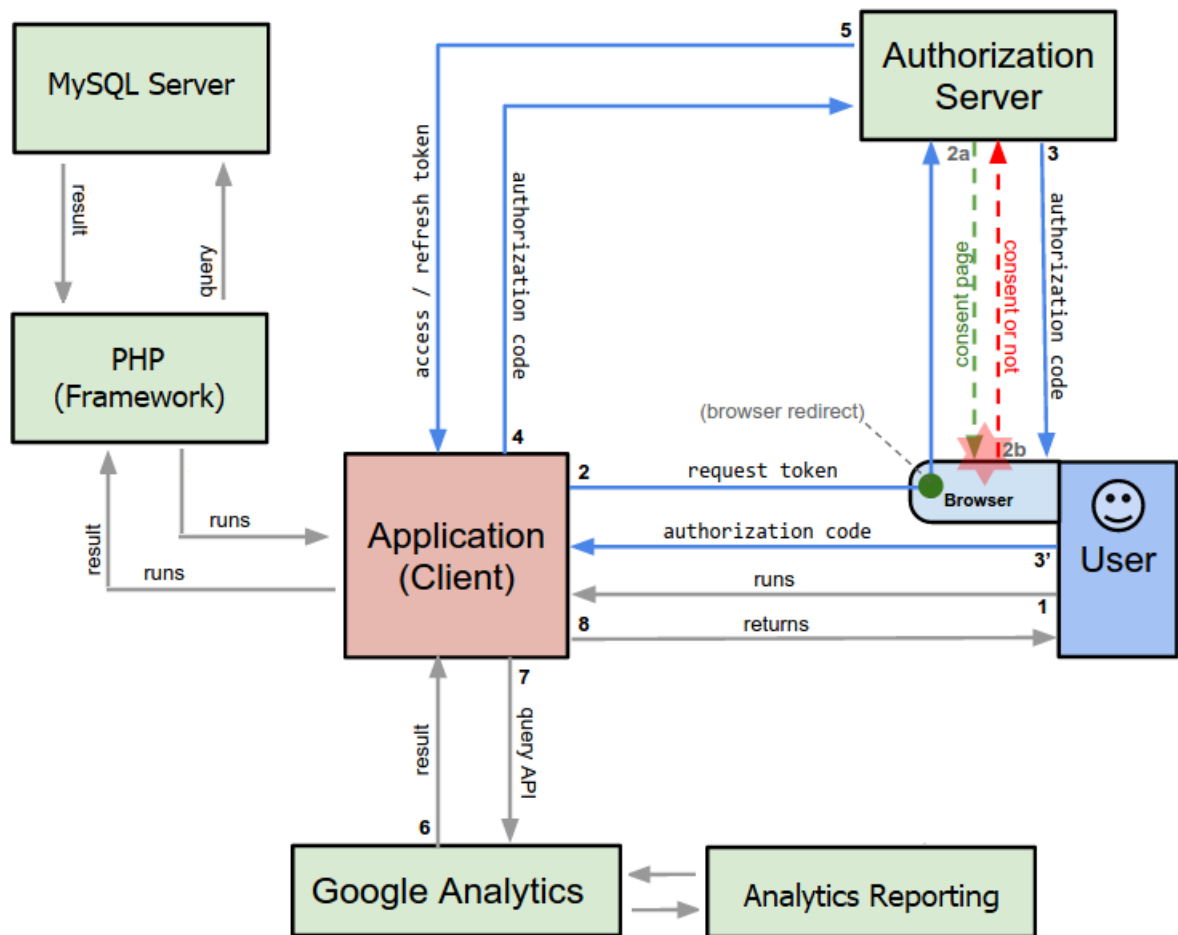


Рисунок 2.1. Процедура отримання інформації через Google API та її запис в СУБД MySQL

У лістингу 2.1 наведено реалізацію методу у визначення кількості переглядів сторінки, використовуючи Google API (Analytics Reporting 4).

```
// діапазон дат
$dateRange = new Google_Service_AnalyticsReporting_DateRange();
$dateRange->setStartDate("2019-09-09");//старт проекту
$dateRange->setEndDate("2020-09-09");

// завдання показника, ga:pageviews
$pageviews = new Google_Service_AnalyticsReporting_Metric();
$pageviews->setExpression("ga:pageviews");
$pageviews->setAlias("pageviews");

// створення фільтру по параметру ga:pagePath
$dimensionFilter =
https://cnap.rada.te.ua/services/list?view=service&id=66
```

Лістинг 2.1. Визначення кількості переглядів через Google API

Як варіант отримані дані можна завантажити в форматі JSON в деякий файл (в цьому прикладі `/assets/ga_pageviews.json`). Для цього потрібно написати відповідну функцію, наприклад, `printResultInFile`.

```
printResultInFile($response);
```

Потім можна написати невеликий скрипт, наприклад, `get_views.php`. Він буде знаходити в файлі кількість переглядів для певної сторінки знаючи її шлях.

Цей скрипт буде складатися з однієї функції (лістинг 2.2):

```
<?php
function getViews($path) {
    $result = 0;
    $items = json_decode(file_get_contents('ga_pageviews.json'));
    foreach($items as $item) {
        if ($item->{'ga:pagePath'} === $path) {
            $result = $result + (int)$item->{'pageviews'};
        }
    }
    return $result;
}
```

Лістинг 2.2. Знаходження кількості переглядів за окремими сторінками

Тепер, щоб отримати дані, необхідно просто викликати функцію `getViews`:

```
echo getViews('/html-and-css/css-selector')
```

2.4. Приклад ідентифікації інтервальної моделі ефективності функціонування інформаційних веб-ресурсів

Застосування описаної вище процедури розглянемо на прикладі веб-ресурсу центру надання адміністративних послуг щодо екологічної експертизи Тернопільської міської ради. Використовуючи GOOGLE ANALITIC та Google API (Analytics Reporting 4), було проведено збір

статистики активності частини користувачів, яка стосується послуг даної експертизи. Дослідження проводилися протягом року. На рис. 2.2 наведено фрагмент графіка активності.

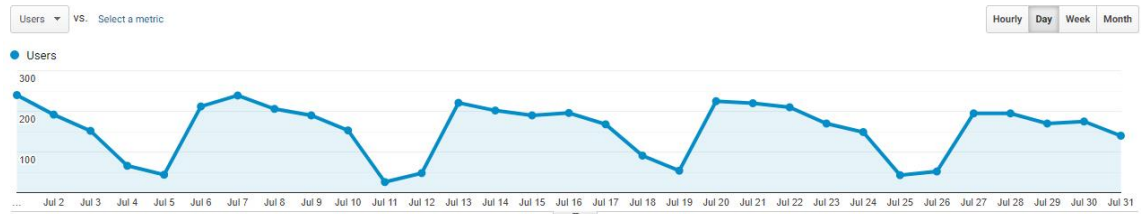


Рисунок 2.2. Фрагмент графіка активності.

У цьому фрагменті бачимо, що активність користувачів має циклічність з періодом в тиждень, що є абсолютно природнім для даного веб-ресурсу. Спираючись на такі спостереження, було прийнято рішення визначити інтервали активності для кожного дня тижня протягом року. Результати зазначених досліджень наведено на рисунку 2.3 та таблиці 2.1.

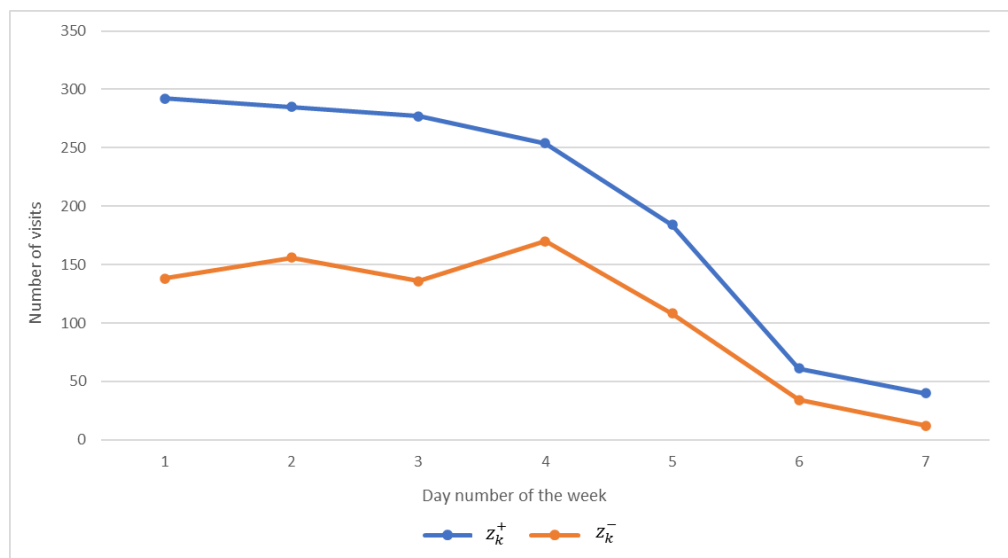


Рисунок 2.3. Інтервали активності для кожного дня тижня протягом року.

Як відомо, процедура параметричної ідентифікації ІДМ є підзадачею задачі структурної ідентифікації [89]. Саме тому для побудови вищезгаданої моделі також потрібно розв'язати і задачу структурної ідентифікації. Для цього використаємо відомий метод структурної ідентифікації ІДМ, який

детально описано у праці [89]. Зауважимо, що цей метод побудовано на біологічних аспектах харчової поведінки колонії медоносних бджіл у дикій природі. Для побудови ІДМ було використано усереднені експериментальні дані активності користувачів зазначеного ресурсу, наведені у таблиці 2.1.

Таблиця 2.1. Дані інтервалу активності.

День тижня	k	z_k^+	z_k^-
1	0	292	138
2	1	285	156
3	2	277	136
4	3	254	170
5	4	184	108
6	5	61	34
7	5	40	12

Під час першого обчислювального експерименту здійснили спробу побудувати ІДМ у вигляді лінійної різницевої схеми. У ході дослідження було отримано структуру математичної моделі у такому загальному вигляді:

$$\hat{v}_k = g_0 + g_1 \cdot \hat{v}_{k-2} + g_2 \cdot \hat{v}_{k-1}, \quad k = 2, \dots, 6. \quad (2.16)$$

Водночас, на етапі параметричної ідентифікації із використанням описаного у 4 розділі цієї статті методу параметричної ідентифікації було отримано такі оцінки параметрів моделі: $g_0 = 0.6644$, $g_1 = -0.6178$, $g_2 = 1.5105$.

Результати моделювання з використанням отриманої математичної моделі разом з експериментальними даними показані на рисунку 2.4. Зокрема, розрахована нижня межа інтервалу $\hat{v}_k^-$ позначається сірою пунктирною лінією, а верхня межа $\hat{v}_k^+$ жовтою пунктирною лінією.

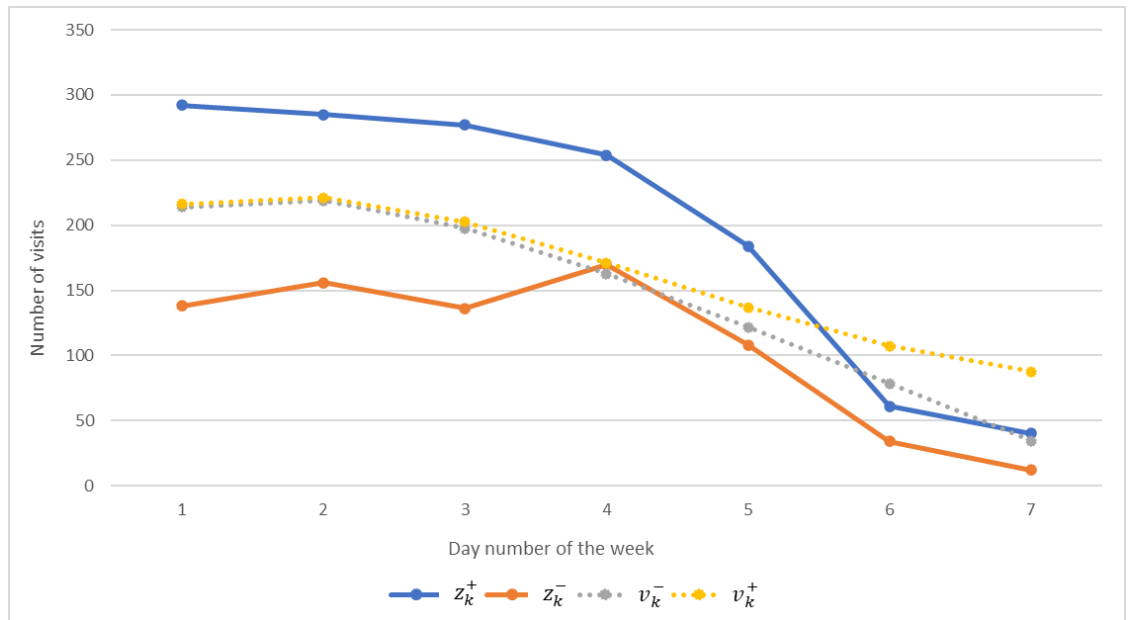


Рисунок 2.4. Прогнозовані значення активності користувача на вказаному ресурсі отримані з використанням інтервальної дискретної моделі (ІДМ) (2.16).

Результати моделювання з використанням отриманої математичної моделі разом із експериментальними даними показані на рисунку 3. Зокрема, розрахована нижня межа інтервалу $\hat{v}_k^-$ позначається сірою пунктирною лінією, а верхня межа $-\hat{v}_k^+$ жовтою пунктирною лінією.

На рис. 2.4 видно, що зазначена модель є досить неточною. Зокрема, для дискрет $k = 3, 5, 6$ (для 4, 6, 7 дня) не виконуються умови (2.5). Зауважимо, що за адекватної структури ІДМ ці умови повинні виконуватися на множині усіх дискрет.

Тому у наступних дослідженнях було прийнято рішення побудувати ІДМ для прогнозування активності користувачів зазначеного ресурсу у нелінійному вигляді. Для цього, під час розв'язування задачі структурної ідентифікації моделі, множину структурних елементів було розширено нелінійними елементами. Таким чином у результаті другого обчислювально

експерименту було отримано таку структуру ІДМ для прогнозування активності користувачів зазначеного ресурсу:

$$\hat{v}_k = g_0 + g_1 \cdot \frac{1}{\hat{v}_{k-1}} + g_2 \cdot \hat{v}_{k-1} \cdot \hat{v}_{k-1} + g_3 \cdot \frac{\hat{v}_{k-1}}{\hat{v}_{k-2}}, k = 2, \dots, 6. \quad (2.17)$$

При цьому на етапі параметричної ідентифікації було отримано такі оцінки параметрів моделі (2.17): $g_0 = -0.2954$, $g_1 = 0.1549$, $g_2 = 0.0042$, $g_3 = -3.3365$.

Як видно з рисунку 2.5, знайдена у ході виконання другого обчислювального експерименту ІДМ також не володіє гарантованими прогностичними властивостями повною мірою. Зокрема, для дискрет $k=3, 5, 6$ не виконуються умови (2.5). Однак, також варто зауважити, що зазначена структура моделі є значно точнішою від структури моделі (2.16) і є близькою до адекватної.

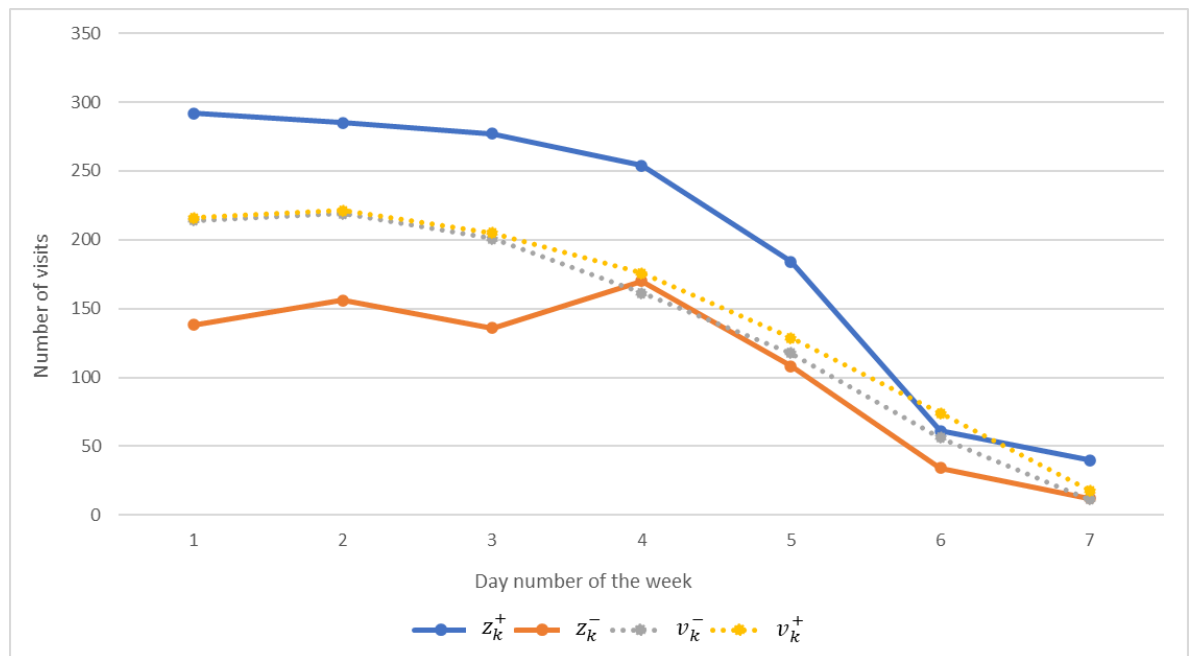


Рисунок 2.5. Прогнозовані значення активності користувача на зазначеному ресурсі отримані за допомогою ІДМ (2.17).

Виходячи з цього, ході третього обчислювального експерименту було прийнято рішення залишити структуру моделі незмінною (2.17) та ще раз

провести процедуру параметричної ідентифікації. При цьому також було прийнято рішення розширити множину початкових умов наступним чином: $k=0\dots 2$.

Так були отримані інші оцінки параметрів моделі (2.17) для прогнозування дискретних значень з числами $k = 3, \dots, 6$ для активності користувачів на зазначеному ресурсі, зокрема: $g_0 = -0.2657$, $g_1 = 0.3150$, $g_2 = 0.0041$, $g_3 = -3.2707$. Математична модель має такий вигляд:

$$\hat{v}_k = -0.2657 + 0.3150 \cdot \frac{1}{\hat{v}_{k-1}} + 0.0041 \cdot \hat{v}_{k-1} \cdot \hat{v}_{k-1} - 3.2707 \cdot \frac{\hat{v}_{k-1}}{\hat{v}_{k-2}}, \quad k = 3, \dots, 6. \quad (2.18)$$

Прогностичні властивості ІДМ за результатами третього обчислювального експерименту наведено на рисунку 2.6. Параметри на малюнку 5 показують результати застосування моделі для прогнозування активності користувачів. Як бачимо, прогнозована активність повністю включена в інтервал спостережень і свідчить про те, що модель є адекватною. Це також забезпечує гарантовані прогностні властивості моделі, що полягають у виконанні умов (2.5) на множині всіх дискретних значень модельованої характеристики.

Варто зазначити, що для прогнозування активності користувача досить встановити значення активності як у перші 3 дні.

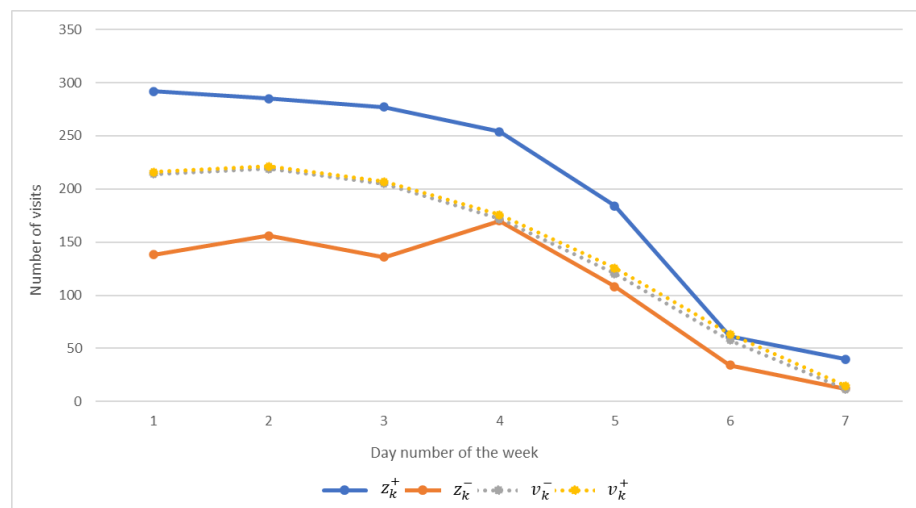


Рисунок 2.6. Результати прогнозування активності користувачів протягом тижня.

Отже, отримана математична модель активності користувачів веб-ресурсу центру надання адміністративних послуг щодо екологічної експертизи Тернопільської міської ради може бути поширена на інші подібні веб-ресурси і вимагає незначного налаштування. Застосування такої моделі дає можливість оптимізувати бізнес-процеси персоналу, планувати протягом тижня щоденну завантаженість виконавців послуг та оптимально розподіляти ресурси організації.

Висновки до розділу 2

1. Запропоновано підхід до моделювання базового показника оцінки ефективності функціонування інформаційних веб-ресурсів, а саме активність користувачів. Він дозволяє врахувати характерні для нелінійних систем особливості – залежність збурень від поточних значень показника, циклічність процесів.

2. Отримано математичну модель у вигляді інтервального різницевого оператора, що дає можливість спрогнозувати активність користувачів веб-ресурсу протягом тижня. Дана модель також має певний ступінь універсалізму і може бути використана для подібних веб-ресурсів із мінімальним налаштуванням. Метод ідентифікації параметрів моделі реалізовано за допомогою АБК.

3. Практичне застосування отриманої моделі в складі веб-ресурсу дає можливість оптимізувати бізнес-процеси в організації, виявити вузькі місця при наданні послуг, планувати протягом тижня щоденну завантаженість виконавців послуг, оптимально розподіляти ресурси організації та завантаженість працівників центру у наданні адміністративних послуг.

РОЗДІЛ 3

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ПРИКЛАДНИХ ПРОГРАМНИХ СИСТЕМ ДЛЯ НАДАННЯ АДМІНІСТРАТИВНИХ ПОСЛУГ

Створення цифрових систем надання адміністративних послуг сьогодні є одним із основних світових трендів. В багатьох працях та на веб-ресурсах, що стосуються даного питання, подано основні вимоги і конкретні рекомендації щодо реалізації зазначених систем. Для цього переважно створюються мобільні додатки, які наближають до сервісів велику кількість користувачів. Водночас такі системи не зменшують часу надання послуг, оскільки, вимагають достатньої обізнаності користувача з особливостями отримання тих чи інших послуг.

Універсальним розв'язком цієї проблеми є створення простих інтелектуальних модулів, які б слугували порадиниками користувачу при отриманні тієї чи іншої послуги. Основою для цих модулів можуть слугувати знання експертів з цієї чи іншої галузі надання послуг.

Для створення системи надання адміністративних послуг потрібно визначити чіткий набір правил, які при запитах користувача могли б сформувати рекомендації щодо прискорення виконання тих чи інших етапів і процедур їх надання.

У цьому розділі запропоновано метод оцінки подібності адміністративних послуг, який ґрунтується на кластеризації документів, що оперують поняттям відстані між об'єктами (реквізитами), які формують їх структуру, вдосконалений метод пошуку оптимальних наборів документів у наведеному графі при відповідних заданих обмеженнях у частині адаптації “алгоритму мурашиних колоній” в заданій предметній області.

Вдосконалена схема реалізації пошуку послуг (документів) у вигляді дерева в частині реалізації алгоритму видалення послуг з документами, що дублюються. Описано основні особливості реалізації інтелектуального модуля як надбудову до сучасних рішень в інформаційних системах надання адміністративних послуг.

3.1 Математичне забезпечення інтелектуального модуля

Сьогодні центри надання адміністративних послуг є мережею з великою кількістю вузлів (послуг) та зв'язків (документів), що робить процес класифікації та пошуку відповідних послуг в системі досить складним. Кожен користувач, вибираючи відповідну послугу, зіштовхується із проблемою – які саме види послуг із отриманого списку вибрати і в якій послідовності необхідно опрацьовувати документи для найповнішого формування відповідного пакету документів та найшвидшої оптимальної обробки інформаційних запитів. На рисунку 1 представлено приклад, де зображено процес отримання видачі дозволу на викиди забруднюючих речовин в атмосферне повітря.

Процес аналізу структури документів та послідовності реалізації відповідних кроків є досить складним, вимагає застосування методів інтелектуального аналізу даних.

Описана проблема може бути визначена як проблема пошуку оптимальних наборів документів, яка може бути транспонована в проблему пошуку оптимальних наборів документів в направленому графі при відповідно заданих обмеженнях.

Для пошуку оптимальних наборів документів при виборі відповідних адміністративних послуг пропонується використати підхід, який базується на основі «мурашиних алгоритмів» [102,103].

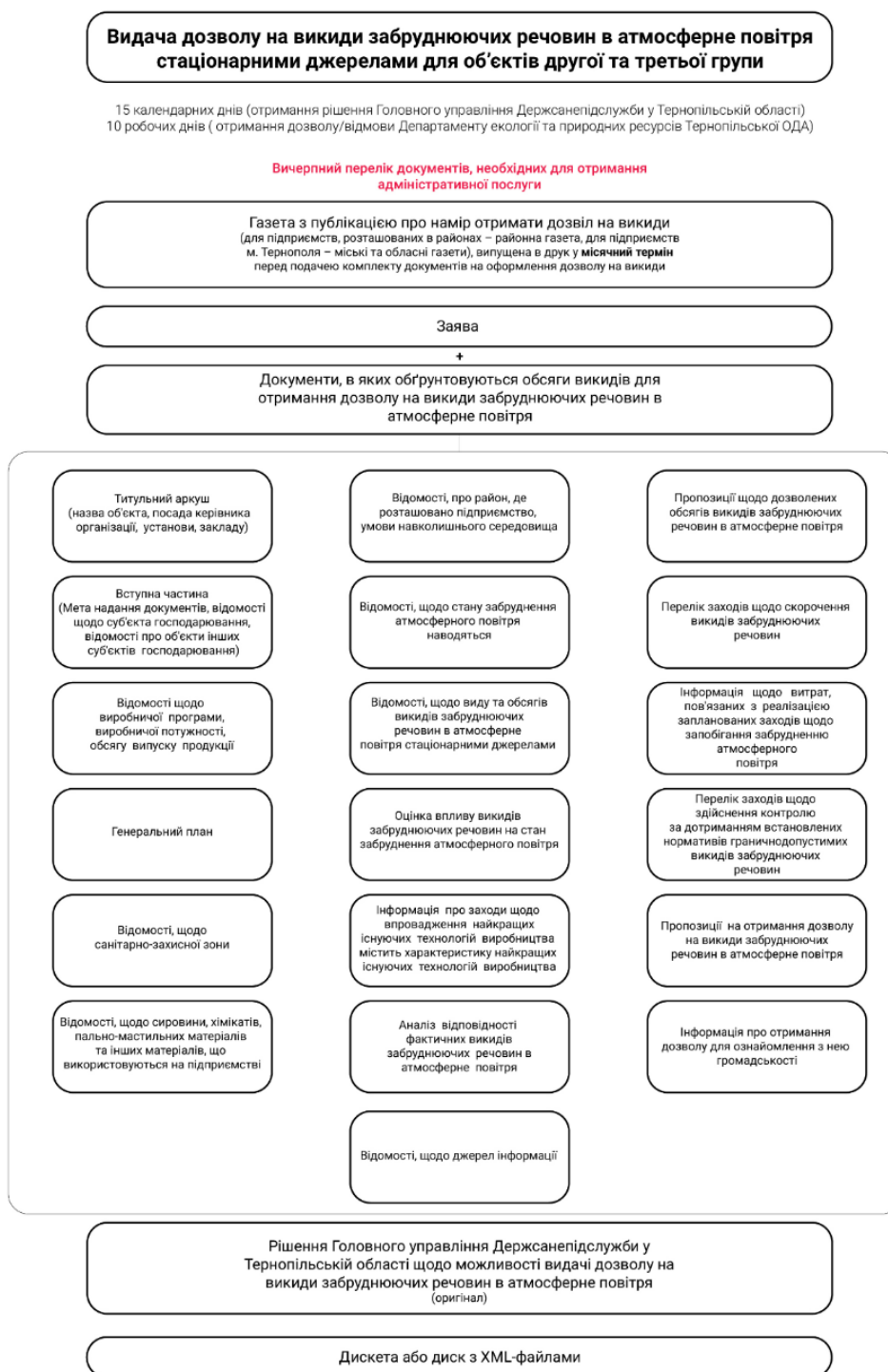


Рисунок 3.1. Приклад реалізації схеми отримання видачі дозволу на викиди забруднюючих речовин в атмосферне повітря

Під контентною повнотою C деякого набору послуг d представимо деяку величину, яка є сумою контентних повнот усіх послуг, що входять в базовий набір адміністративних послуг:

$$C_p^n = \sum_{j=1}^{n-1} c_j, \quad (3.1)$$

де n - кількість послуг (вершин графу), c_j – контентна повнота деякої адміністративної послуги.

Нехай в результаті формування відповідного інформаційного запиту Q сформовано структурну модель у вигляді графу G , що відображає взаємозв'язки між документами, які отримані в результаті відповідного пошуку. Необхідно сформувати такий набір послуг, який охоплював би максимальну кількість документів, контентна повнота якого була б максимальною, а контентна повнота документів, що не відповідають запиту – мінімальною.

В загальному випадку задача може бути представлена наступним чином:

$$\text{Search } C \text{ with } n \rightarrow \max, C_k^n \rightarrow \max, C_{k_{\text{uninformative}}}^n \rightarrow \min \quad (3.2)$$

Відношення (3.2) є класичним представленням пошуку оптимальних шляхів обходу в направлених графах. Розглянемо специфіку застосування методу мурашиних колоній в рамках досліджуваної предметної області.

Опис методу. Моделювання поведінки мурах пов'язане з розподілом феромону на шляху – ребрі графу. Ймовірність того, що мураха піде по конкретному ребру пропорційна кількості феромону на цьому ребрі, а кількість феромону, що відкладається на шляху, пропорційна «відстані» між двома документами. В якості міри відстані може бути обрана будь-яка із загальновідомих [103]. Таким чином, чим «блищими» будуть два документи, тим більше феромону буде відкладено на ребрі, що їх з'єднує, і, відповідно, тим більше мурах буде рухатися по даному маршруту. Для попередження знаходження локально оптимальних рішень необхідно додати зворотній зв'язок у вигляді випарування феромону. Це дасть змогу отримати стійке оптимальне рішення.

Розглянемо правила поведінки мурах:

1. Мурахи можуть рухатися по ребрах декілька разів, але це може спонукати до дублювання фрагментів шляху. В спірних моментах (вузлах, документах) мурахи повинні віддавати перевагу тим ребрам, які вони ще не відвідали. Позначимо через $M_{i,ant}$, перелік документів, які ще не проходила комаха ant , що знаходиться на i -й послугі.

2. Видимість визначається як евристичне бажання комахи відвідати документ j , якщо він формує послугу i . Даний показник обчислюємо як обернено пропорційну величину до значення «відстанні» між документами i відповідних послуг. Для цього використаємо критерій χ^2 Пірсона [3]:

$$\chi^2 = n_1 n_2 \sum_{i=1}^k \frac{1}{m_{1,i} + m_{2,i}} \left(\frac{m_{1,i}}{n_1} - \frac{m_{2,i}}{n_2} \right)^2, \quad (3.3)$$

де n_1 - об'єм першого документу, n_2 - об'єм другого документу, $m_{1,i}$ - частота i -го признаку в першому послугі, $m_{2,i}$ - частота i -го признаку в другій послугі.

3. «Нюх» комах визначається їх здатністю відчувати запах феромону, що підтверджує бажання відвідати документ j із послуги i на основі досвіду інших мурах. Кількість феромону на ребрі (i, j) в певний момент часу t позначимо як $W_{i,j}(t)$.

Таким чином, на основі описаних правил запишемо відношення ймовірності переходу ant –ої мурахи з послуги i в послугу j :

$$\begin{cases} Probability_{ij,ant}(t) = \frac{[\tau_{ij}(t)]^\alpha * [\eta_{ij}]^\beta}{\sum_{l=1}^{J_{i,ant}} [\tau_{il}(t)]^\alpha * [\eta_{il}]^\beta}, j \in J_{i,ant}, \\ Probability_{ij,ant}(t) = 0, j \notin J_{i,ant}. \end{cases} \quad (3.4)$$

Параметри α і β визначаються експериментально.

Величина феромону, що відкладе мураха ant на ребрі (i, j) , перейшовши з послуги i на послугу j , визначається таким чином:

$$\Delta W_{ij,ant}(t) = \begin{cases} F_{P_{ant}(t)}, & (i,j) \in P_{ant}(t), \\ 0, & (i,j) \notin P_{ant}(t), \end{cases} \quad (3.5)$$

де $P_{ant}(t)$ - шлях, що пройшла мураха ant на момент часу t , $F_{P_{ant}(t)}$ - числова величина оптимальності шляху p .

Числова величина оптимальності шляху розраховується на основі оптимізаційної функції

$$F_p = n_p C_p v_p, \quad (3.6)$$

де n_p - кількість документів, що входять в маршрут p , C_p - контентна повнота (1) маршруту p , v_p - середня швидкість зміни контентної повноти із врахуванням порядкового номеру документу в маршруті p , що обчислюється за формулою:

$$v_p = \frac{\sum_{l=1}^{n_p-1} \frac{(l+1)i}{\sum_{j=1}^{l+1} i_j}}{n_p - 1} \quad (3.7)$$

Випаровування феромону обчислюється за наступною формулою:

$$\begin{aligned} W_{ij}(t+1) &= (1-p) * W_{ij}(t) + \Delta W_{ij}(t), \\ \Delta \tau_{ij}(t) &= \sum_{k=1}^m \Delta W_{ij,k}(t), \end{aligned} \quad (3.8)$$

де $p \in [0;1]$ – коефіцієнт випаровування, m – кількість мурах (послуг).

В результаті ми отримуємо модифікований метод мурашинних колоній, його адаптивне представлення та застосування в сфері пошуку документів і відповідних адміністративних послуг в процесі функціонування інформаційних систем ЦНАП.

В процесі аналізу результатів пошуку адміністративних послуг з використанням інформаційних систем ЦНАП виникає проблема автоматичного фільтрування подібних послуг та відповідних документів. Це дозволить значно пришвидшити результати такої обробки [104-107].

Поняття подібності адміністративних послуг можна описати через кластеризацію документів, що оперують поняттям відстані між об'єктами (реквізитами), які формують структуру цих документів. Відстань між такими об'єктами можна знаходити за допомогою критерію χ^2 Пірсона, який описаний відношенням (3.3). Відповідно, коефіцієнт подібності можна обчислювати як:

$$S_{p_1, p_2} = 1/\chi^2 \quad (3.9)$$

Під подібністю адміністративних послуг будемо розуміти ступінь дублювання даних, що вони містять:

$$\text{Similarity}(\text{Service}_1, \text{Service}_2) = \text{Data}_1 \cup \text{Data}_2 \quad (3.10)$$

Під індексом подібності будемо розуміти числову характеристику подібності, причому повному дублюванню даних буде відповідати значення 1, відповідно 0 - послугам, що повністю відрізняються.

Нехай S_1 і S_2 – дві адміністративні послуги, для яких необхідно визначити їх подібність, o_1 і o_2 – набір унікальних ознак відповідних послуг [4]. Під унікальними ознаками мається на увазі набір ознак, що не повторюються в межах однієї послуги. В якості таких ознак будемо використовувати елементи інформаційного наповнення, а саме текстові описи, документи. Загальний набір ознак O (3.11) утворюється шляхом об'єднання набору ознак послуг, що порівнюються. Ознаки, що повторюються, також видаляються з набору:

$$O = o_1 \cup o_2, \text{count}(O) \leq \text{count}(o_1) + \text{count}(o_2) \quad (3.11)$$

Коефіцієнт дублювання даних i -ї послуги в j -й будемо обчислювати за допомогою відношення

$$W_{s_i(s_j)} = \frac{\text{count}(o_i) + \text{count}(o_j) - \text{count}(O)}{\text{count}(o_i)} \quad (3.12)$$

Загальний коефіцієнт даних, що дублюється в послугах, p_1 і p_2 будемо обчислювати за допомогою виразу

$$W_{s_i, s_j} = \frac{\text{count}(o_i) + \text{count}(o_j) - \text{count}(O)}{\text{count}(o_i) + \text{count}(o_j)} \quad (3.13)$$

а коефіцієнт подібності відповідно як

$$\text{Similarity}_{s_i, s_j} = 2W_{s_i, s_j}, \text{ Similarity}_{i, j} \in [0, 1], \quad (3.14)$$

Відповідно відстань між послугами може бути обчислена за допомогою виразу:

$$D_{p_i, p_j} = \frac{1}{\text{Similarity}_{s_i, s_j}} \quad (3.15)$$

Таким чином, коефіцієнт подібності враховує невідповідність наборів документів, а відстань між послугами є певною мірою обмеженою, так як коефіцієнт подібності може змінюватися лише в діапазоні від 0 до 1. Отже, у такий спосіб вирішується проблема розмірності відстані і подібності.

3.2. Алгоритмічне забезпечення реалізації інтелектуального модуля

Алгоритм видалення послуг з документами, що дублюються, представлено на рисунку 3.2. Дані методи можуть бути реалізовані як надбудова до сучасних рішень і ефективно використовуватися в інформаційних системах надання адміністративних послуг.

На першому кроці користувач вводить пошуковий запит, система видає результати – набір послуг S.

На другому кроці аналізуються перші n документів для адміністративних послуг: вони розбиваються на блоки, оцінюються за допомогою регресійної моделі, інформаційний шум «відкидається». На виході отримуємо n документів, які складаються лише з основних документів для потрібної послуги.

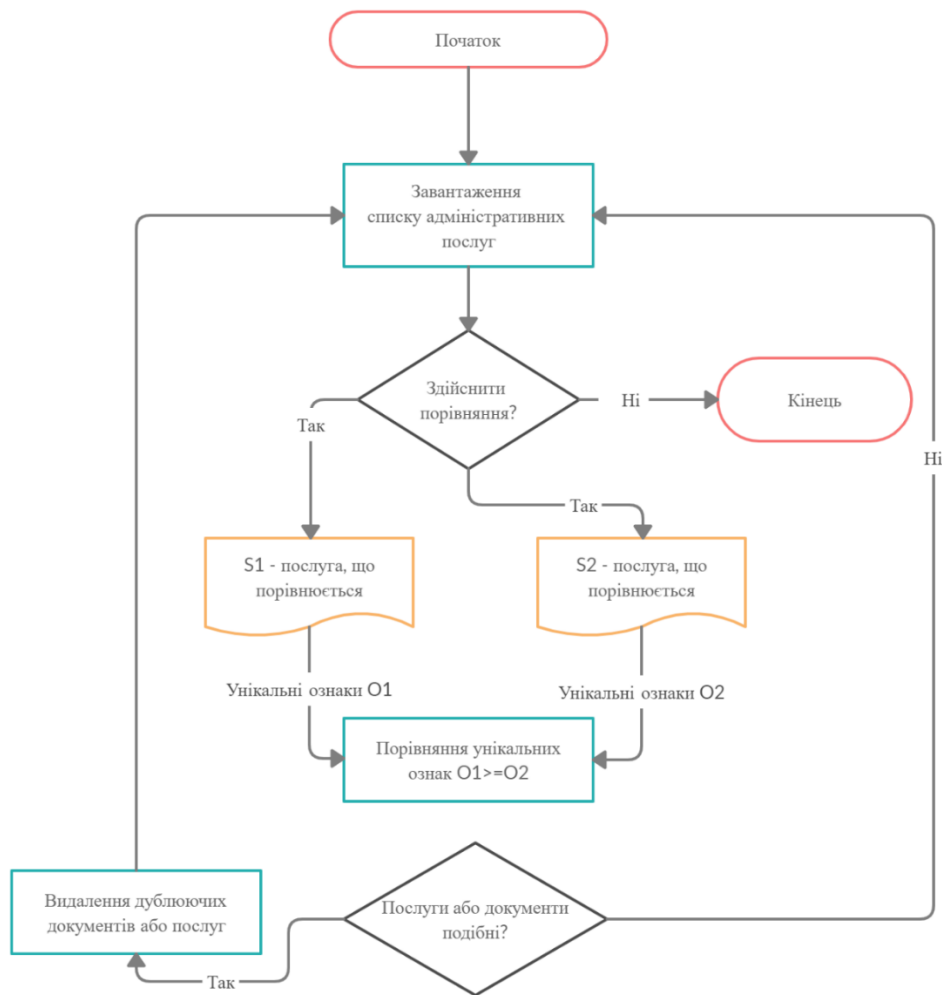


Рисунок 3.2. Схема видалення дублюючих документів та послуг в процесі формування пошуково запиту на адміністративні послуги

На третьому кроці обчислюються коефіцієнти подібності між документами. Всі документи, що дублюються в рамках послуг, видаляються з результуючого набору (в залежності від методу видалення).

На виході отримуємо набір послуг з документами інформація в яких не дублюється.

Дані методи та алгоритми можуть бути реалізовані як надбудова до сучасних рішень і ефективно використовуватися у власних програмних розробках.

Схема реалізації інтелектуального модуля. Інтелектуальний модуль виконує деревовидний пошук документів – від загального до конкретного [108-110]. Проводиться заглиблення у нижчі рівні здійснюваного пошуку. На рисунку 3.3 представлено схему реалізації пошуку документів у вигляді дерева.

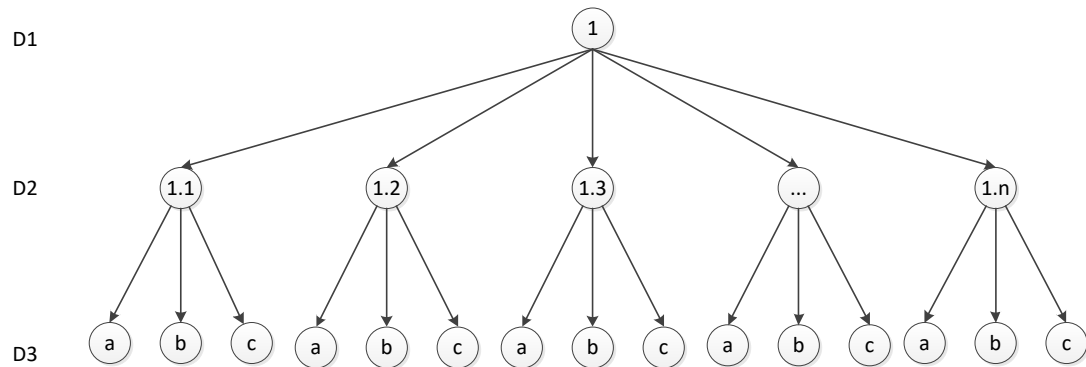


Рисунок 3.3. Схема пошуку документів у вигляді деревовидного представлення

Для того, щоб із пошукових результатів обрати потрібний варіант, потрібно виконати наступну послідовність кроків D, де:

D_1 – результати пошуку в конкретному розділі адміністративних послуг;

D_2 – знайдена інформація під час пошуку;

D_3 – на третьому рівні знаходяться: а) розділи за якими переходить користувач; б) документи або текст, які відносяться до результатів пошуку; с) відсоток адекватності пошуку.

Спочатку користувач заповнює поля пошуку, вибирає відповідні атрибути і при натисненні на кнопку «Знайти адміністративну послугу» відбувається відправлення даних на сервер, де відбувається формування словників вибраних елементів із надісланих даних. На основі згенерованої інформації відбувається безпосереднє формування запиту, його аналіз та

розбиття на деталізовані структури. На кожному вузлі відбувається порівняння виконаного запиту із статистичною інформацією по запропонованому алгоритму [111-121].

Після виконання пошуку на вузлах через менеджер бази даних документів здійснюється повернення набору даних, де вже виконується об'єднання результатів і відповідно надсилаються агреговані результати до сторінки формування запиту на адміністративну послугу. Там користувач може у зручному вигляді переглядати знайдені результати та виконувати подальші маніпуляції з потрібними документами [122].

3.3. Особливості реалізації програмного забезпечення

Дисертаційне дослідження виконувалося на прикладі Центру надання адміністративних послуг міста Тернополя. З метою підвищення ефективності діяльності інформаційної системи ЦНАП та визначення специфіки реалізації програмного забезпечення, яке реалізовувалося, виникла необхідність конкретизації бізнес-процесів даної предметної області та її представлення у формі відповідних описових структур. Розглянемо детальніше основні процеси та виділимо проблемні моменти їх програмної реалізації.

Виходячи з отриманої інформації, є можливість конкретизувати предметну область, визначитись з переліком необхідних робіт/завдань, спланувати подальші дії.

Враховуючи те, що користувачам потрібно мати можливість віддалено переглядати інформацію, задавати питання та виконувати певні дії, система, яка розроблювалась, є веб-орієнтованим додатком.

Веб-додаток включає усі необхідні дані, які можуть бути корисними для громадян. Зокрема, представлено структуру центру із вказанням залів, посад, номерів телефонів. Також веб-додаток містить положення про Центр

надання адміністративних послуг, регламент роботи, закони України. На окремих сторінках відображено графіки роботи, контактну інформацію, телефони, інші методи зв'язку (пошта, скайп), карту, транспортне сполучення із переліком найближчих зупинок та парковок [122-129].

Оскільки в центрі надання послуг часто з'являється нова інформація, яку швидко потрібно донести до користувачів, на сайті створена стрічка новин. На окрему сторінку надходитимуть останні новини із сайту міської ради.

Довідник послуг групується за категоріями та суб'єктами надання. Також є можливість переглядати повний список послуг з можливістю швидкого пошуку та групування по залах, де ці послуги надаються. На сторінці кожної послуги надається основна інформація, така як назва, шифр, зал, категорія, суб'єкт надання, графік, строки та результати надання, вичерпний перелік документів, а також прикріплені файли із інформаційною карткою, зразками заяв.

Кожен клієнт може заповнити документ із формою заяви і відправити його на перевірку онлайн, не витрачаючи часу на візит у центр. До подачі запиту можна перейти напряму із сторінки послуги (пункт «Перевірка документів онлайн»), в такому випадку форма запиту автоматично заповниться даними послуги. Діаграма подачі документів на перевірку зображена на рисунку 3.4.

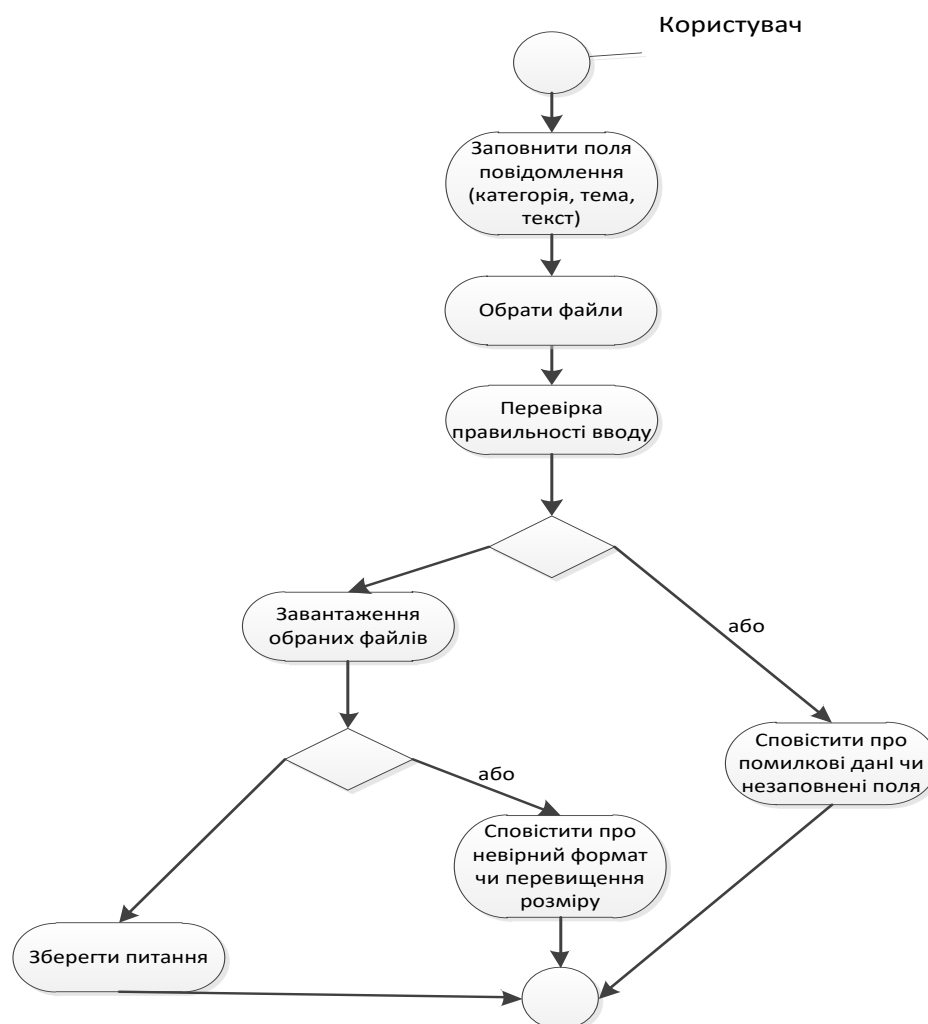


Рисунок 3.4. Діаграма подачі документів на перевірку

Подача і перегляд питань відбувається у кабінеті користувача. Окрім цього, у своєму кабінеті користувач має можливість переглянути список частих запитань або ввести свої персональні дані, які можуть бути використані для запису на прийом чи перегляду наданих послуг. В такому варіанті нема потреби кожен раз повторно вводити своє ім'я та номер телефону. Діаграма входу в кабінет зображена на рис. 3.5.

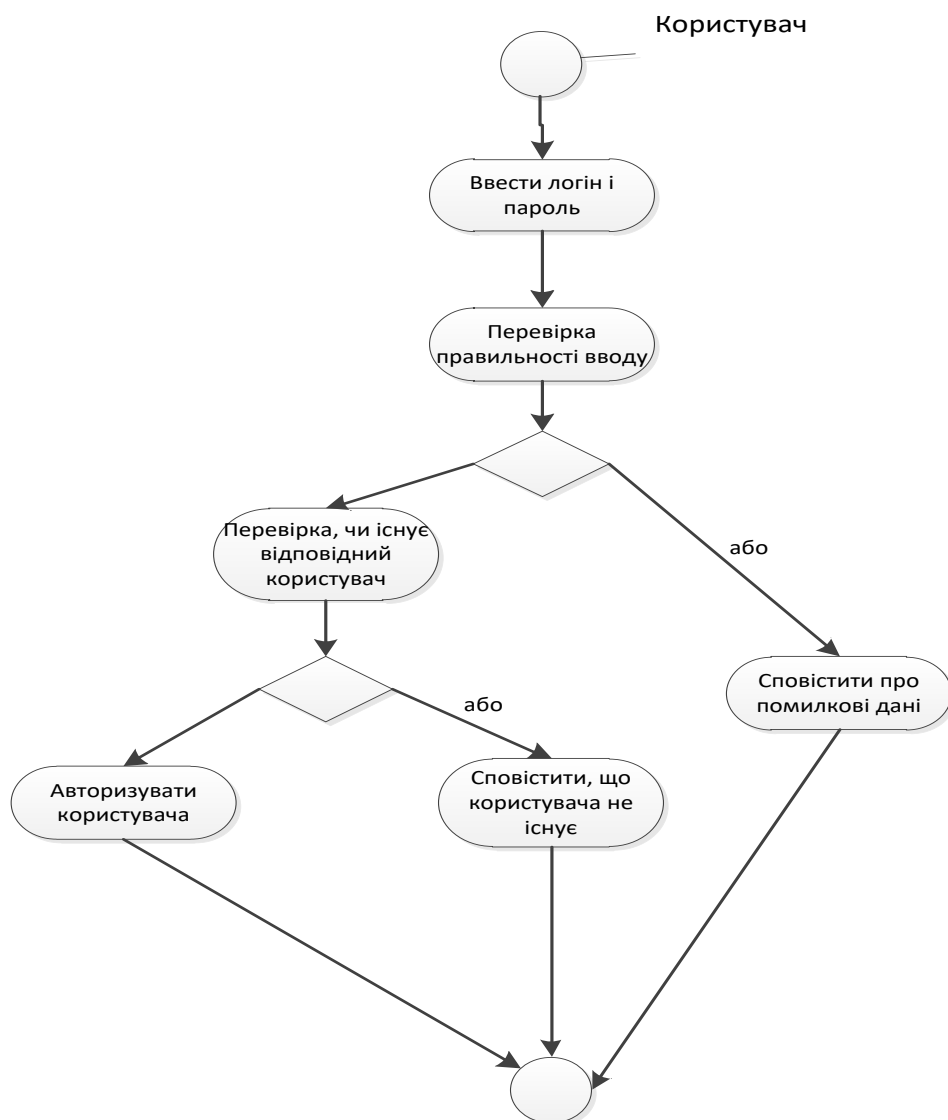


Рисунок 3.5. Діаграма входу у кабінет

Для того, щоб отримати логін і пароль для доступу в особистий кабінет, новому користувачу потрібно зареєструватися (рис. 3.6).

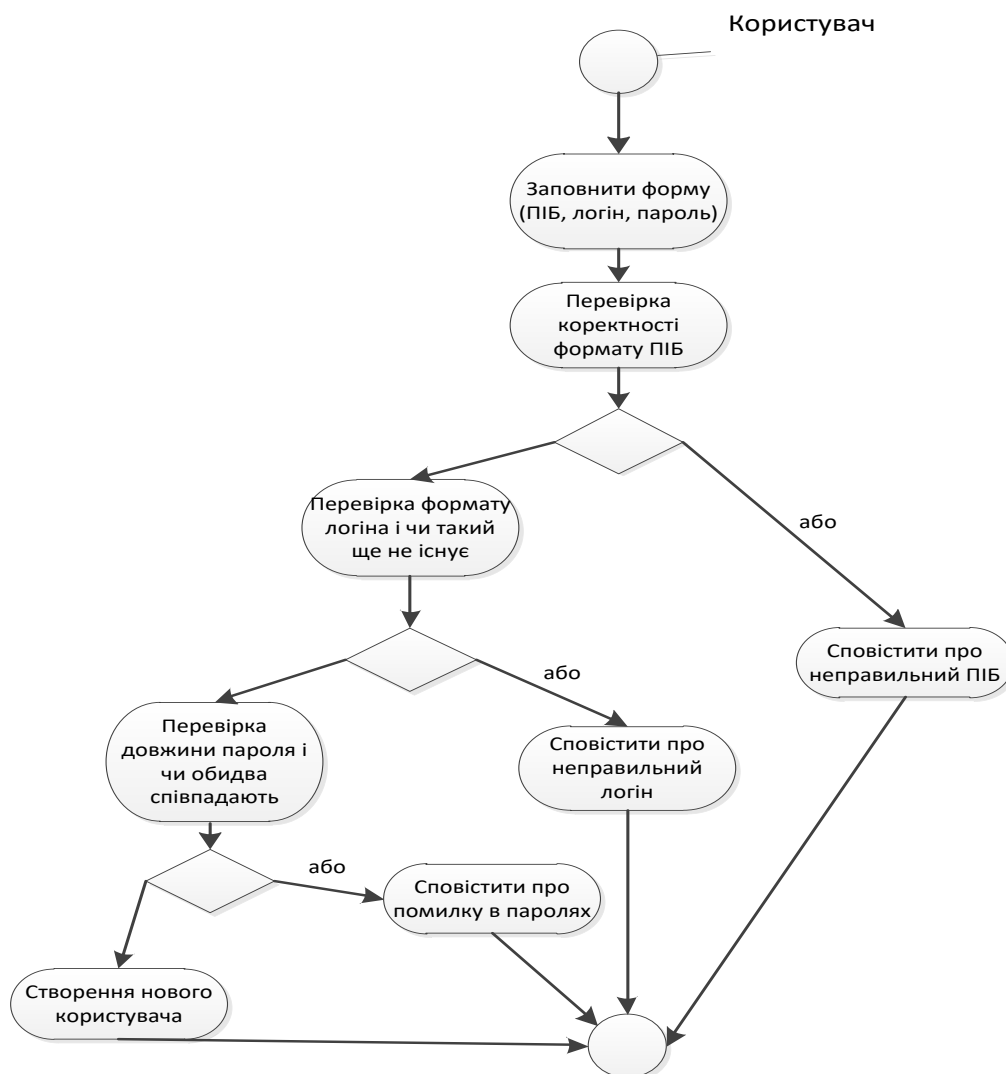


Рисунок 3.6. Діаграма реєстрації користувача

Для запису на прийом в електронну чергу потрібно обрати послугу чи зал надання, день та годину. Для реєстрації вказати своє ПІБ, номер телефону, кількість питань для вирішення, підтвердити згоду на збір та обробку персональних даних. Після успішної реєстрації у чергу клієнт отримує можливість роздрукувати чек, який зможе показати при отриманні послуги. Дата реєстрації вважається зайнятою і стає недоступною для інших користувачів. Діаграма запису на прийом зображена на рис. 3.7.

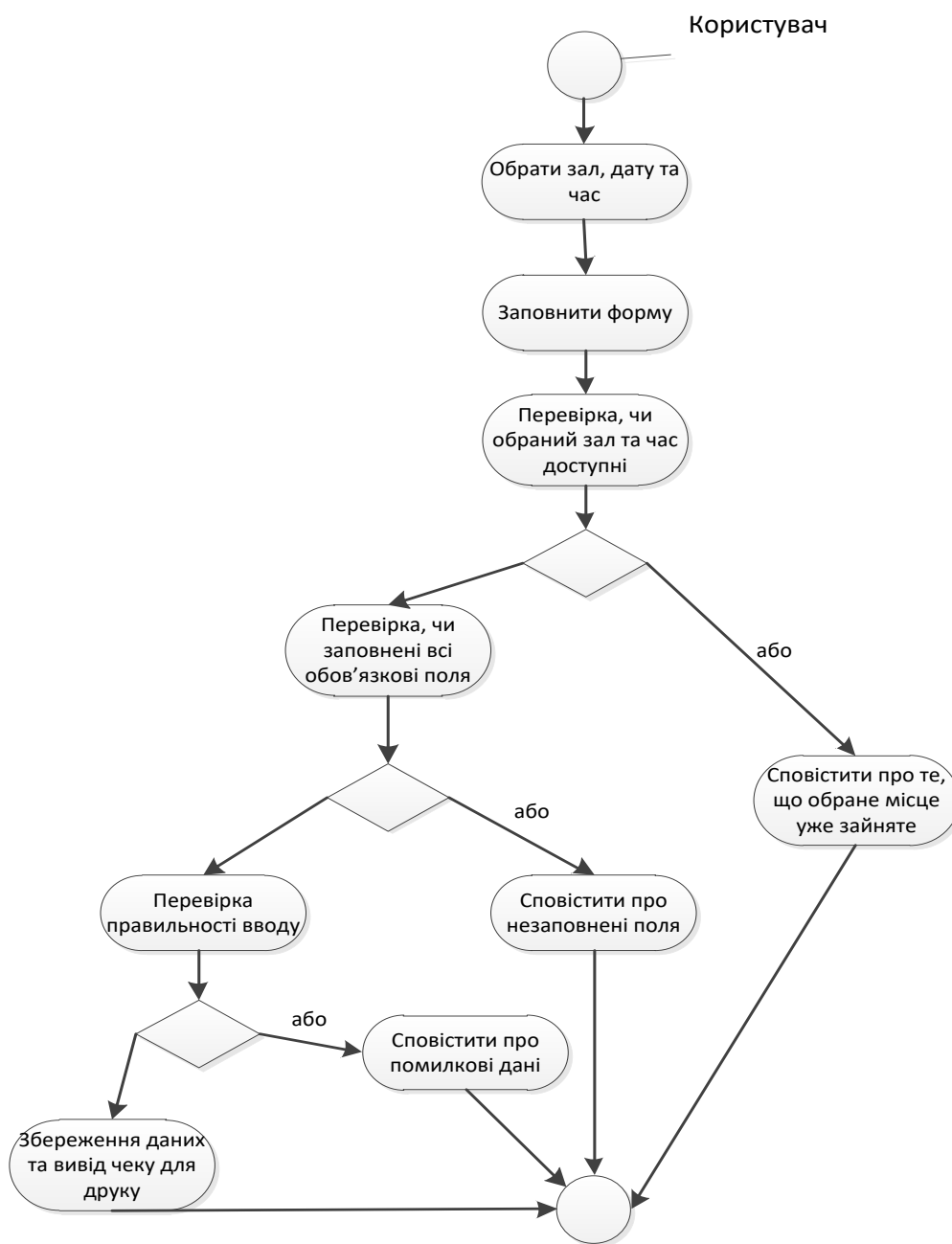


Рисунок 3.7. Діаграма запису на прийом

Для людей, які уже були на прийомі і подали документи для певної послуги, передбачений розділ в кабінеті користувача «Надані послуги», де можна переглянути свої послуги, стан їх виконання (задоволено чи відмовлено), примітки, зміщення термінів виконання. Для того, щоб подати цю інформацію на сайті, потрібно налагодити регулярну синхронізацію із локальною базою ЦНАПу, в якій зберігаються детальні дані з усіх наданих

послуг. Діаграма входу до перегляду стану опрацювання послуг зображена на рисунку 3.8.

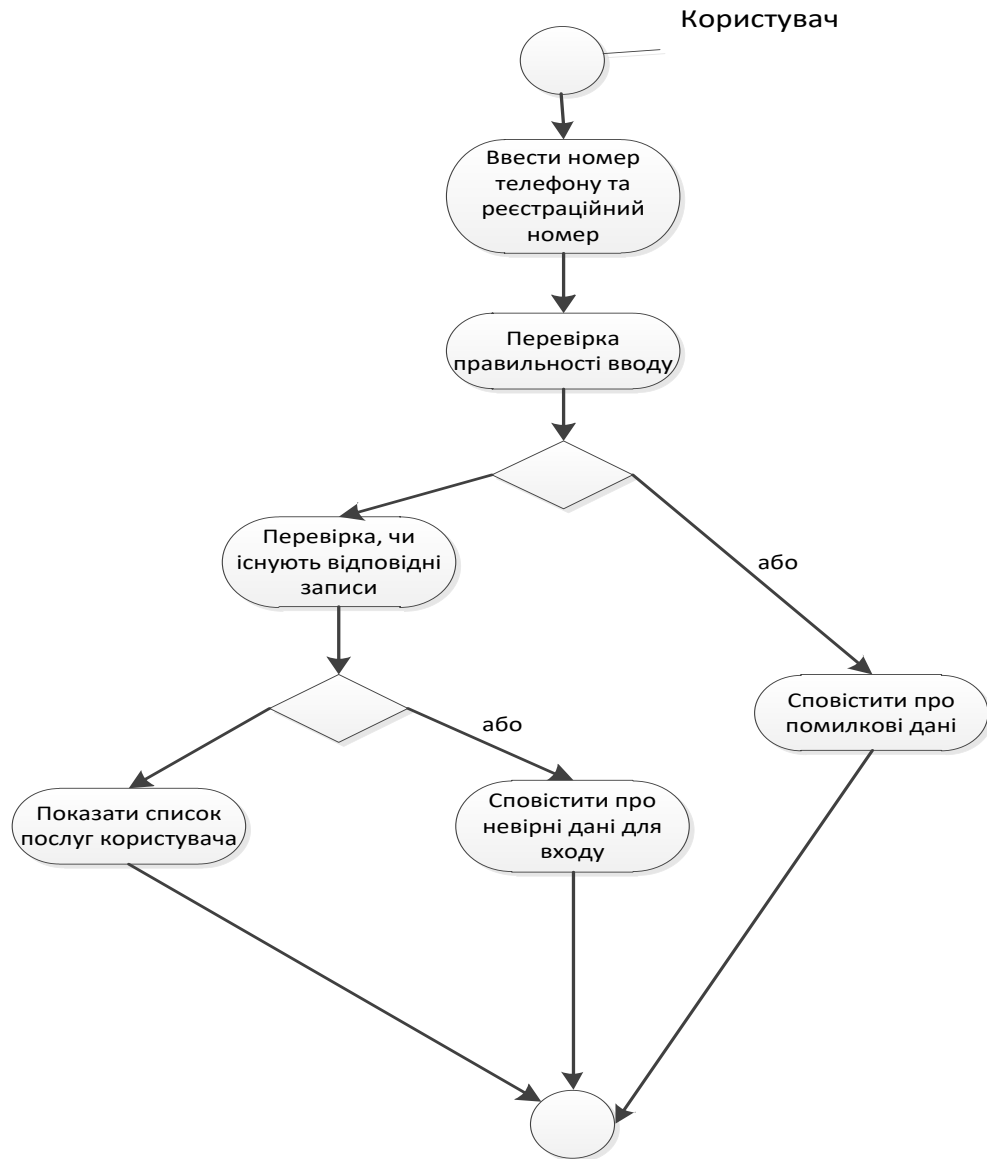


Рисунок 3.8. Діаграма перегляду наданих послуг

Також надаватиметься можливість залишити до кожної із отриманих послуг позитивний чи негативний відгук, що дасть змогу адміністрації центру перевіряти якість надання послуг та виявляти проблемні моменти, які потребують покращення. Діаграма відправки відгуку зображена на рисунку 3.9.

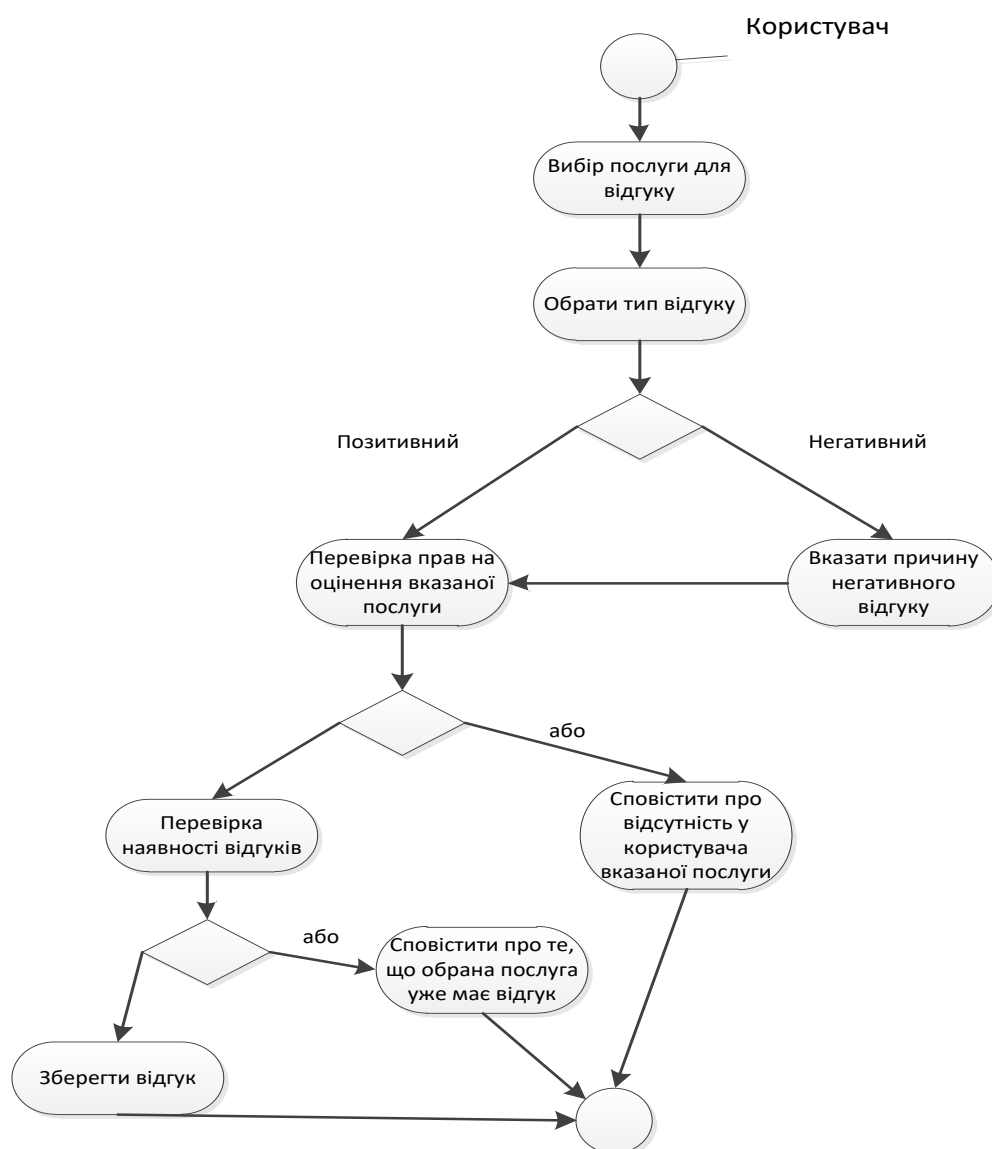


Рисунок 3.9. Діаграма відправки відгуку

Окрім персональних відгуків на сайті потрібно проводити загальні опитування, наприклад, щодо оцінки роботи центру, зручності сайту та інші.

У користувачів часто виникають питання на які на сайті або немає повної відповіді, або користувач просто не зміг знайти потрібної інформації. В таких випадках буде діяти онлайн консультація, де працівник центру в режимі реального часу допомагатиме відвідувачам сайту із вирішенням їх питань.

Окрім загальнодоступних можливостей на сайті повинен функціонувати окремий адміністративний інтерфейс для працівників центру. Працівник зможе додавати новини, редагувати інформацію, створювати нові опитування. Можна буде вказати графік запису в електронну чергу, а також вихідні та святкові дні на які запис не ведеться. Тут же доступні для перегляду усі записи в електронну чергу із можливістю переглядати список лише для обраної дати та залу, щоб могли швидко пересвідчитись, що на прийом прийшла саме та людина, яка зареєструвалась на сайті. Також можна анулювати підозрілі реєстрації (в яких є очевидним вказання невірної ПІБ чи телефону), а усі неактуальні дані, дата яких уже минула, повинні автоматично видалятися, щоб не нагромаджувати застарілої інформації та зменшити кількість персональних даних у системі. Для наглядності на списках послуг та залів буде вказана кількість позитивних та негативних відгуків, які можна переглянути на окремій сторінці.

Всі функції адміністративного інтерфейсу доступні за гнучкою системою розмежування прав доступу, де можна створювати нових користувачів та надавати їм потрібні дозволи на виконання тих чи інших дій.

В загальному вигляді адміністративна частина представлена як стандартизований інтерфейс, що забезпечує зручність та пришвидшує навчання персоналу для роботи з ним. Діаграма додавання нового запису (новини, послуги і т.д.) в систему зображена на рисунку 3.10.

Виходячи із наведених вище діаграм (3.4-3.10) виникла проблема інтеграції інтелектуального модуля в існуючу інформаційну систему ЦНАП.

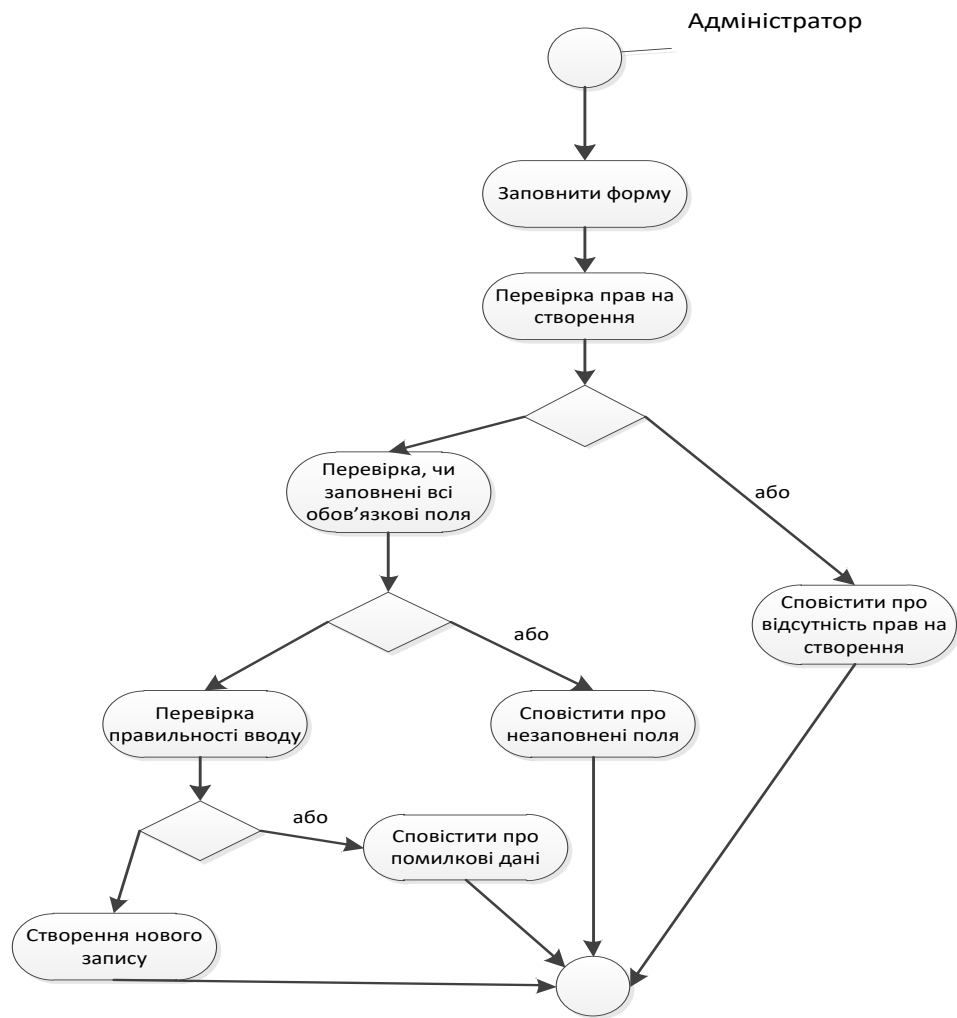


Рисунок 3.10. Діаграма додавання нового запису

На рис. 3.11 представлено приклад реалізації алгоритму оцінки подібності адміністративних послуг та приклад візуального представлення отриманих результатів.

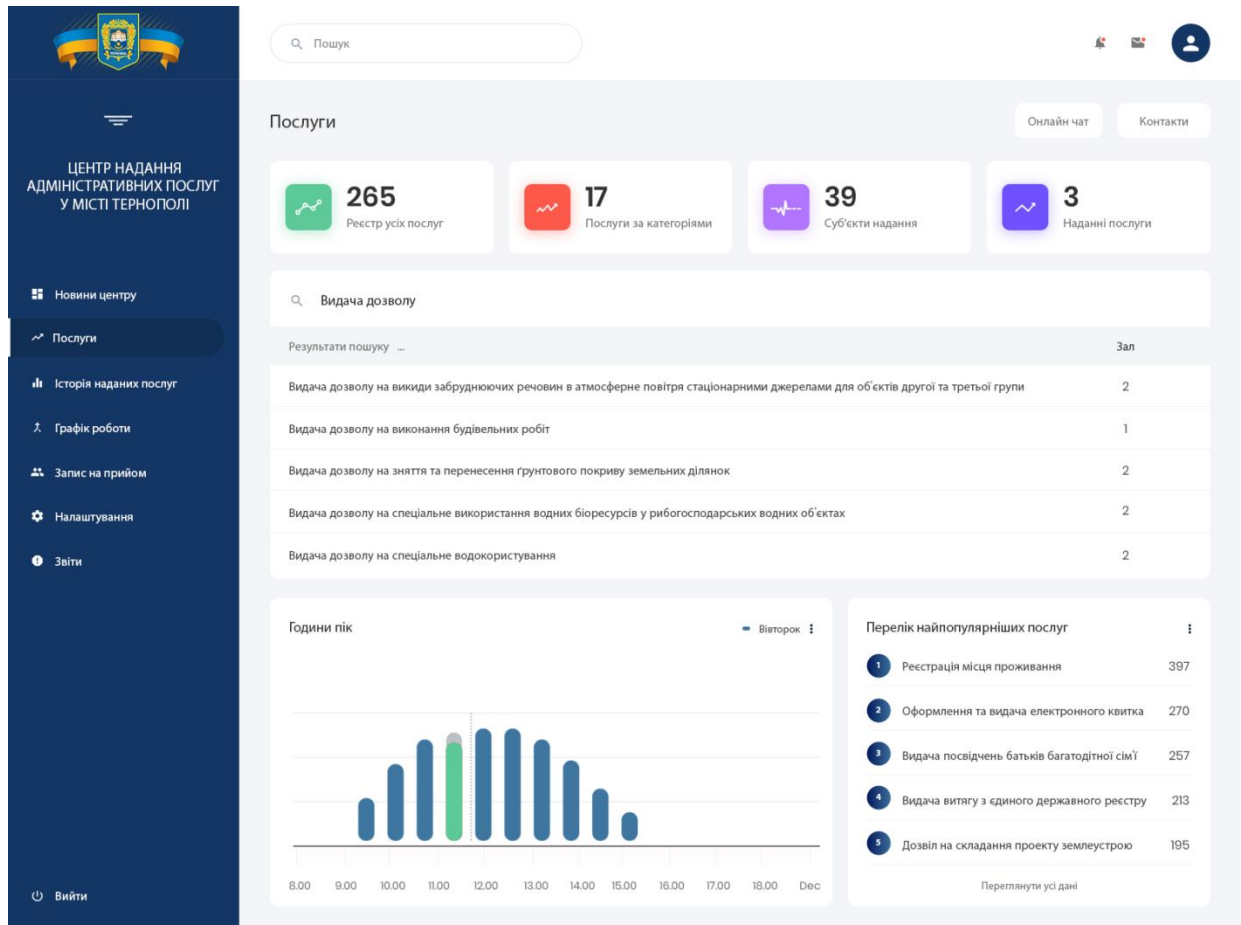


Рисунок 3.11. Приклад реалізації оцінки подібності адміністративних послуг

На рисунку 3.12 представлено приклад формування структури документів для адміністративної послуги «Видача дозволу на викиди забруднюючих речовин в атмосферне повітря стаціонарними джерелами для об'єктів другої та третьої групи».

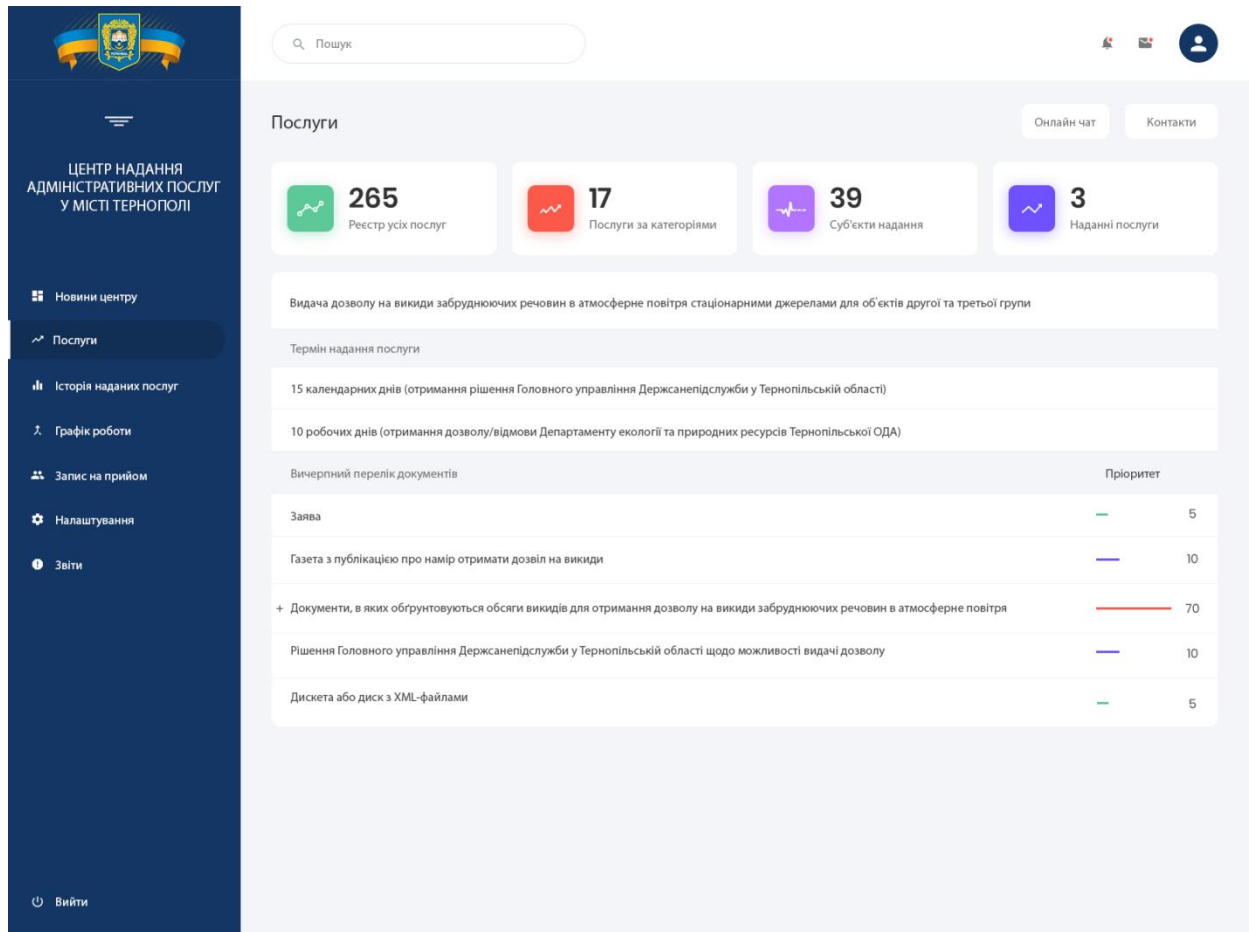


Рисунок 3.12. Інтерфейс інтелектуального модуля обробки інформації для системи центру надання адміністративних послуг

Отримана структура документів сформована з використанням інтелектуального модуля обробки інформації і є більш інформативною та простішою для пересічних користувачів.

Висновки до розділу 3

1. Розроблений метод оцінки подібності адміністративних послуг, який ґрунтується на кластеризації документів, що оперують поняттям відстані між об'єктами (реквізитами), які формують їх структуру.

2. Вдосконалено метод пошуку оптимальних наборів документів у наведеному графі при відповідних заданих обмеженнях в частині адаптації “алгоритму мурашиних колоній” заданої предметної області.

3. Вдосконалено схему реалізації пошуку послуг (документів) у вигляді дерева в частині реалізації алгоритму видалення послуг з документами, що дублюються.

4. Реалізовано інтелектуальний модуль обробки інформації, який базується на використанні методу пошуку оптимальних наборів документів шляхом адаптації “алгоритму мурашиних колоній” в рамках досліджуваної предметної області. Запропоновані методи та засоби можуть бути реалізовані як надбудова до сучасних рішень і ефективно використовуватися в інформаційних системах надання адміністративних послуг.

РОЗДІЛ 4

ПРИКЛАДНА ПРОГРАМНА СИСТЕМА ДЛЯ МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЇ ПРОЦЕСІВ ДЛЯ ЦЕНТРУ НАДАННЯ АДМІНІСТРАТИВНИХ ПОСЛУГ

Практична реалізація надання усіх видів адміністративних послуг з метою отримання ефективних результатів для користувачів, пов'язана з використанням сучасних інформаційних технологій. Важливим елементом автоматизації бізнес-процесів надання адміністративних послуг є розробка інтелектуальних модулів та засобів їх імплементації.

Інтелектуалізація процесів надання адміністративних послуг дає можливість користувачам швидко реалізовувати свої інформаційно-комунікаційні запити.

В рамках виконання дисертаційного дослідження реалізовано інтелектуальний модуль обробки інформації про надавані послуги та оптимізації процесів їх надання за часом. Інтелектуальний модуль обробки інформації базується на використанні методу пошуку оптимальних наборів документів шляхом адаптації “алгоритму мурашиних колоній” в рамках досліджуваної предметної області.

Запропоновані в роботі програмні рішення можуть бути реалізовані як надбудова до сучасних інформаційних систем надання адміністративних послуг і успішно використовуватися для підвищення ефективності їх функціонування.

4.1. Архітектура програмної системи

Існують різні види архітектур, які успішно застосовуються для реалізації різноманітних програмних систем. При розробці програмного

забезпечення була використана багаторівнева архітектура. Основними рівнями її є: відображення, бізнес-логіка, доступ до даних. На рисунку 4.1 продемонстровано архітектуру прикладної програмної системи центрів надання адміністративних послуг для рівня муніципалітетів із інтеграцією інтелектуального модуля обробки інформації.

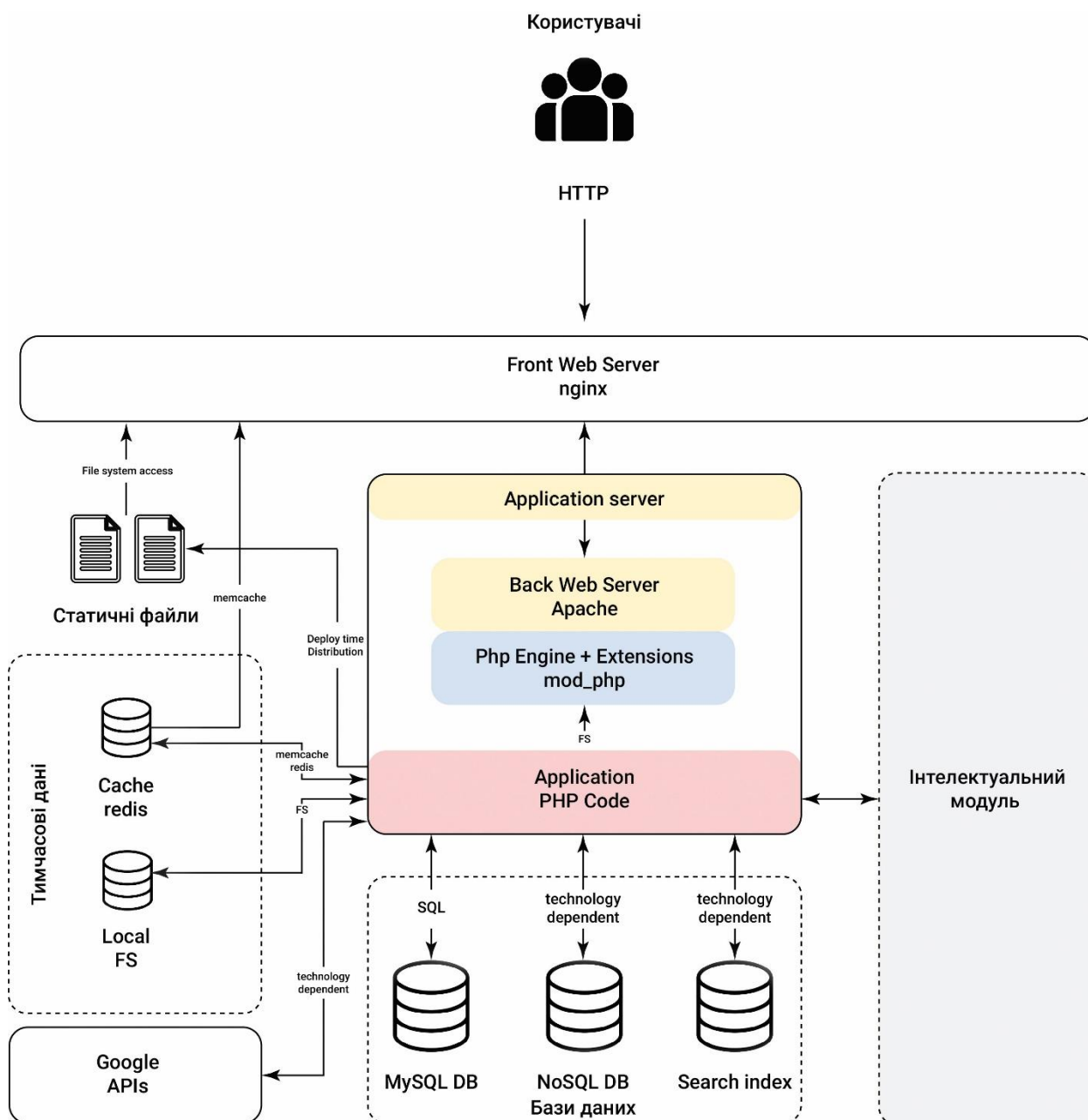


Рисунок 4.1. Архітектура програмної системи для надання адміністративних послуг.

Розглянемо детальніше структуру основних елементів запропонованої архітектури програмного забезпечення: front web server, application server, сервер баз даних, інтелектуальних модуль, зовнішні API, служби кешування.

Веб-сервер (front web server) є сполучною ланкою між веб-додатком і клієнтом. Він приймає запити від клієнта у вигляді набору даних по протоколу HTTP, повертає у відповідь документи на мові HTML і файли, які потрібні додаткові. При цьому документи та інші файли можуть бути як статичними (перебувати в файлової системі сервера), так і динамічними, тобто сформованими програмною частиною. Кожен компонент системи вирішує свою задачу. Завдання front web server обробити вхідний запит і вирішити, що з ним робити далі. На front web server зазвичай ставиться максимально швидкий і універсальний web server. В нашому випадку – nginx. Nginx може обслуговувати десятки тисяч запитів. Додатково налаштовано віддача статичних, зображень, css, JS через Nginx. А це є найпростіший шлях запиту. Наприклад, якщо користувачу потрібне зображення, він надсилає запит на Nginx, Nginx знаходить відповідне зображення і повертає його користувачу.

Якщо звернення йде до динамічного файлу, тобто до коду, який повинен виконувати певну дію, то тут свою роботу розпочинає програмний інтерпретатор (application server).

Програмний інтерпретатор (application server) – це серце системи. За допомогою програмного інтерпретатора виконується код, що містить логіку роботи системи (алгоритми). У веб-середовищі можуть виконуватися додатки, написані на різних мовах програмування. Їх можна розділити на дві групи: компільовані (виконувані як звичайні програми на комп'ютері) і інтерпретовані (що вимагають інтерпретатора для виконання). При розробці веб-додатку використовується інтерпретована мова PHP. Інтерпретовані програми повільніше виконуються, але процес їх розробки значно простіший

і швидший. З огляду на постійно зростаючі вимоги бізнесу на швидкість впровадження змін в програмні продукти (інформаційні системи), швидкий цикл розробки програм стає вирішальним фактором. Крім того, потужність комп'ютерів постійно зростає, дозволяючи до певної межі забувати про продуктивність додатків на догоду зручності і швидкості розробки.

Сервер баз даних виконує функцію зберігання структурованої інформації, а також її видачу за запитами веб-додатків. При цьому між програмною частиною системи і сервером баз даних можуть використовуватися різні інтерфейси. Від сервера баз даних залежать продуктивність програми і функціональність з точки зору даних.

Кожний сучасний веб-додаток використовує одну або декілька баз даних для зберігання інформації. Бази даних надають інструменти для організації, додавання, пошуку, оновлення, видалення та виконання обчислень над даними.

В даній архітектурі використовуються такі бази даних як: SQL, NoSQL, search index [130].

SQL-бази даних зберігають дані в таблицях, які пов'язані між собою загальними ключами.

Більшість SQL погано масштабуються горизонтально, а масштабуватися вертикально можуть тільки до певного моменту.

NoSQL є більш новим набором технологій баз даних. Він був розроблений для обробки дуже великих обсягів інформації, які можуть генеруватися великомасштабними веб-додатками.

Search index – структура даних, яка містить інформацію про документи і використовується в пошукових системах. В архітектурі використовується система повнотекстового пошуку Sphinx (протокол доступу MySQL). Відмінною особливістю є висока швидкість індексації та пошуку, а також інтеграція з існуючими СУБД.

Служба кешування надає просте сховище даних в форматі ключ-значення, яке дозволяє зберігати і шукати інформацію за час, близький до лінійного.

Додаток використовує технологію кешування Redis, щоб зберігати результати обчислень і скористатися ними пізніше з кешу, а не перераховувати їх ще раз. Додаток може кешувати результати запиту в бази даних, результати звернення до зовнішніх служб, HTML для заданого URL-адресу.

Блок стороннього API – це також частина архітектури, до якого звертається наш додаток. В ролі зовнішньої системи, до якої здійснюються запити, виступає Google API (Analytics Reporting 4). Дана система призначена для отримання кількості переглядів визначеної сторінки на сайті. Для зв'язку використовується два протоколи: http і Measurement Protocol.

Вагому роль в архітектурі відіграє інтелектуальний модуль. Завдання цього модуля – пошук оптимальних наборів документів в заданій предметній області.

В основу даного програмного модуля закладено реалізацію бізнес-процесів, які виникають під час опрацювання документів та відповідних адміністративних послуг, а також реалізацію взаємодії в рамках використання методу інтелектуального опрацювання інформації для оцінювання подібності онлайн послуг та метод автоматизованого пошуку оптимальних наборів документів в заданій предметній області.

Інтелектуальний модуль реалізований засобами PHP 7.2 та фреймворку Zend. В додатку А наведено лістинг розробленого модуля інтелектуальної обробки інформації [131,132].

На рисунку 4.2 представлено схему інтеграції інтелектуального модуля.

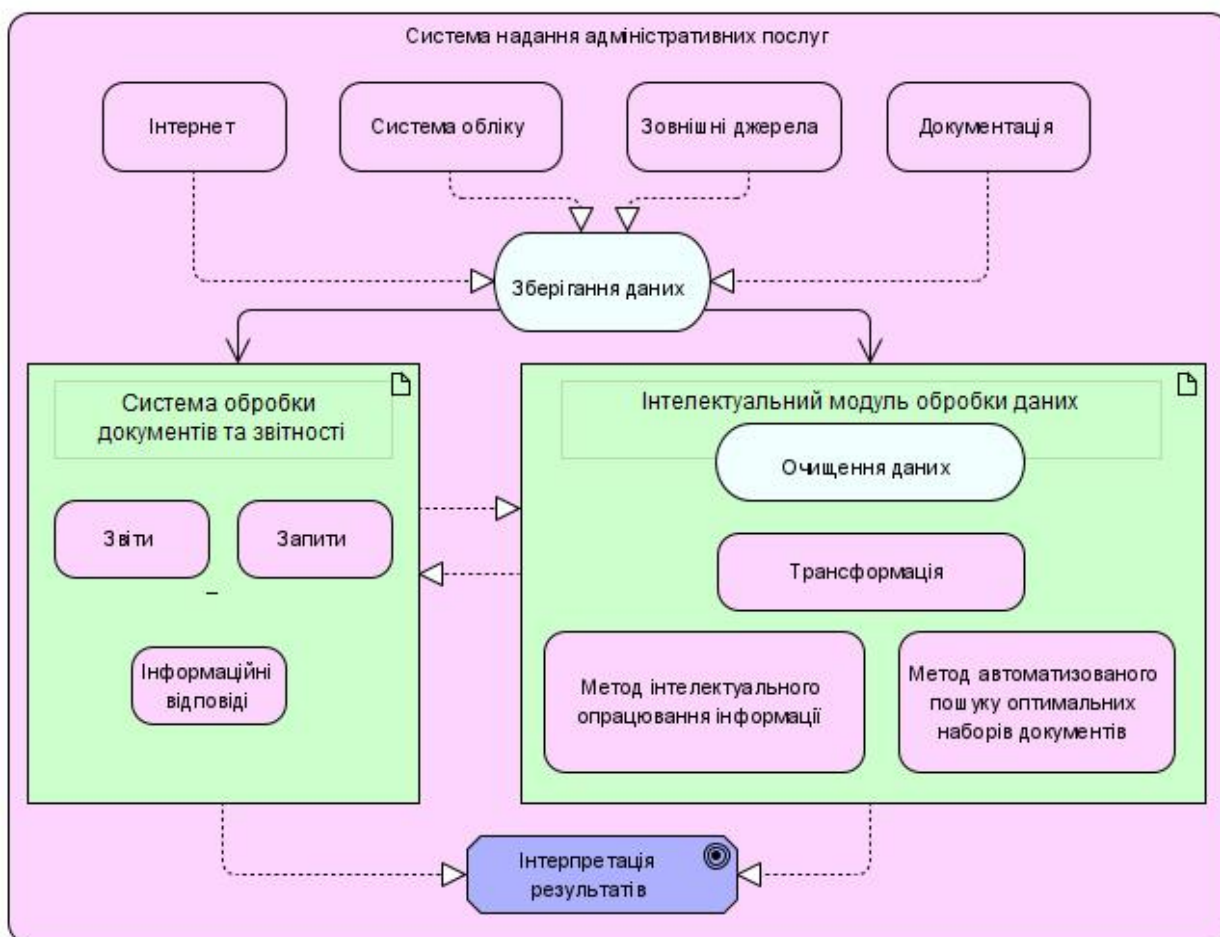


Рисунок 4.2. Схема інтеграції програмного модуля

Розглянемо детальніше програмну реалізацію методів для оцінювання подібності онлайн послуг (лістинг 1), який ґрунтується на пошуку оптимальних наборів документів в заданій предметній області.

```
class DM_Classification extends DM{
    private $alg;
    private $nodes;
    private $edges;
    private $trainingTuples;
    private $targetName;
    private $attrs;
    private $attrVals;
    private $attrType;
    private $clusterSetTuples;
    private $clusterSet;
    private $meanTuples;
    private $inputTuple;
    private $k;
```

```

private $nearestNeighbours;

public function set_C4_5_Attributes($trainingTuples,
$targetName, $attrs, $attrVals){
    $this->trainingTuples= $trainingTuples;
    $this->targetName= $targetName;
    $this->attrs= $attrs;
    $this->attrVals= $attrVals;
}

public function set_NN_Attributes($clusterSetTuples,
$clusterSet, $meanTuples, $inputTuple, $attrs, $attrType, $k){
    $this->clusterSetTuples= $clusterSetTuples;
    $this->clusterSet= $clusterSet;
    $this->meanTuples= $meanTuples;
    $this->inputTuple= $inputTuple;
    $this->attrs= $attrs;
    $this->attrType= $attrType;
    $this->k= $k;
}

public function mine($type){

    $classArr= array(0,1,2);

    if($type == "C4_5"){

        $this->alg= new C4_5($this->trainingTuples,
        $classArr, $this->targetName);
        $this->alg->process_C4_5($this->attrs, $this->attrVals);

        $this->nodes= $this->alg->getNodes();
        $this->edges= $this->alg->getEdges();

        //print_r($this->nodes);
    }
    else if($type == "NN"){

        $this->alg= new K_Nearest_Neighbours();

        $this->nearestNeighbours= $this->alg->get_K_Nearest_Neighbours($this->clusterSetTuples, $this->clusterSet, $this->meanTuples, $this->inputTuple, $this->attrs, $this->attrType, $this->k);
    }
}

public function getNodes(){
    return $this->nodes;
}

public function getEdges(){
    return $this->edges;
}

```

```

        public function getNearestNeighbours(){
            return $this->nearestNeighbours;
        }

        public function generate_decisionTree_ID($time_ID,
$network_ID, $targetName){

            $db= new DB($this->target_db);

            $query = "insert into decisiontree (time_ID,
network_ID, targetAttribute) values ('$time_ID', '$network_ID',
'$targetName')";

            //echo $query;

            $db->insert($query);

            $query= "select decisiontree_ID from decisiontree
order by decisionTree_ID desc limit 0,1";

            //echo $query;

            $decisionTree_ID= $db->retrieveElement($query);
            return $decisionTree_ID;
        }
    }
}

```

Лістинг 4.1. Реалізація пошуку оптимальних наборів документів в заданій предметній області

Лістинг 4.2 відображає програмну інтерпретацію пошуку оптимального набору документів в заданій предметній області, який ґрунтується на кластеризації документів, що оперують поняттями відстані між об'єктами-реквізитами.

```

class DM_Clustering extends DM{
    private $alg;
    private $clusters;
    private $meanTuples;
    private $trainingTuples;
    private $attrs;
    private $attrVals;
    private $attrType;
    private $attrDep;

    public function set_K_Means_Attributes($trainingTuples,
$attrs, $attrVals, $attrType, $attrDep){
        $this->trainingTuples= $trainingTuples;
        $this->attrs= $attrs;
    }
}

```

```

        $this->attrVals= $attrVals;
        $this->attrType= $attrType;
        $this->attrDep= $attrDep;
    }

    public function mine($type){
        if($type == "K-means"){

            $this->alg= new K_means($this->attrs, $this-
>attrVals, $this->attrType, $this->attrDep);
            $this->alg->process_G_Means($this-
>trainingTuples);
            $this->clusters= $this->alg->getClusters();
            $this->meanTuples= $this->alg->getMeanTuples();
        }
    }

    public function getClusters(){
        return $this->clusters;
    }

    public function getMeanTuples(){
        return $this->meanTuples;
    }

    public function generate_clusterset_ID($time_ID,
$network_ID, $type){

        $db= new DB($this->target_db);
        $query = "insert into clusterset (time_ID, network_ID,
type) values ('$time_ID', '$network_ID', '$type')";
        $db->insert($query);
        $query= "select clusterset_ID from clusterset order by
clusterset_ID desc limit 0,1";
        $clusterset_ID= $db->retrieveElement($query);
        return $clusterset_ID;
    }
}

```

Лістинг 4.2. Реалізація пошуку оптимального набору документів, який ґрунтується на кластеризації документів

Розглянемо детальніше особливості впровадження інтелектуального модуля в існуючу інформаційну систему центру надання адміністративних послуг. Цей процес є дуже важливим виходячи з особливостей реалізації основних бізнес процесів. Їх об'єднує структура користувацьких запитів та інформації, яка формує відповідний результат.

Поряд з класичною автоматизацією бізнес-процесів функціонування інформаційної системи надання адміністративних послуг, яка спрощено описано з допомогою діаграм активності (див. розділ 3.3), розглянемо варіант із

підключенням інтелектуального модуля. На рисунку 4.10 представлено діаграму, яка описує зв'язок між користувачами та ЦНАП системою з інтелектуальним модулем.

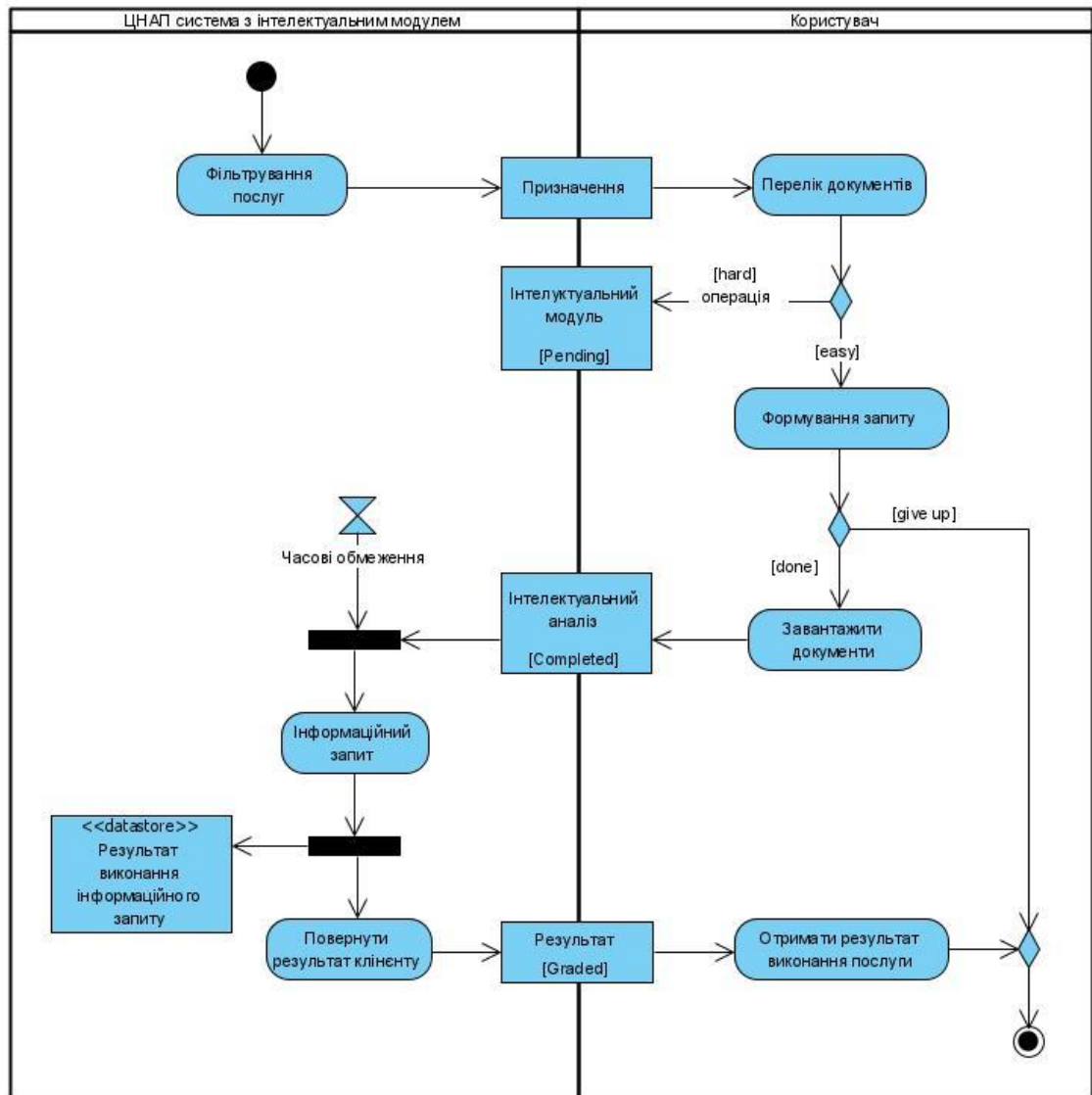


Рисунок 4.3. Діаграма діяльності інтелектуального модуля в загальній системі ЦНАП

Запропонована в даній роботі архітектура, яка лягла в основу програмної системи надання адміністративних послуг, володіє рядом переваг: наявність інтелектуального модуля обробки інформації, можливість

інтеграції в функціонуючі системи, як інтегрований елемент, можливість легкого масштабування виходячи із уточнення функціональних вимог.

4.2. Прикладні аспекти реалізації системи для центру надання адміністративних послуг

Під час реалізації інформаційної системи центру надання адміністративних послуг з підключенням інтелектуального модуля особлива увага приділена розробці відповідного інформаційного забезпечення.

Уся необхідна інформація зберігається у відповідних структурах, які реалізовані в середовищі СУБД MySQL.

Вибір даної СУБД зумовлений виходячи із переваг, описаних у першому розділі даної роботи, а також із вимог забезпечення комплексності функціонування інформаційної системи ЦНАП.

Розглянемо детальніше декілька основних відношень, які використовуються для опису структур документів та відповідних послуг.

DDL-код таблиці, де зберігаються дані послуг представлено у лістингу 4.3. Серед основних атрибутів потрібно відзначити наступні: `name` – назва послуги, `department_id` та `category_id` – ідентифікатор відповідного залу та прив'язаної до нього категорії.

```
CREATE TABLE IF NOT EXISTS `cnap_office_services` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `department_id` int(11) unsigned NOT NULL DEFAULT '0',
  `category_id` int(11) unsigned NOT NULL DEFAULT '0',
  `hall_id` int(11) unsigned NOT NULL DEFAULT '0',
  `code` char(50) NOT NULL,
  `name` varchar(500) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `description` text NOT NULL,
  `description_period` text NOT NULL,
  `description_result` text NOT NULL,
  `description_documents` text NOT NULL,
  `description_files` text NOT NULL,
  `duration_time` tinyint(3) unsigned NOT NULL,
  `derzhposluga` char(250) NOT NULL,
  `published` tinyint(1) NOT NULL DEFAULT '1',
```

```

`ordering` int(11) NOT NULL DEFAULT '0',
`created` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
`created_by` int(11) unsigned NOT NULL DEFAULT '0',
`created_by_alias` char(100) NOT NULL DEFAULT '',
`modified` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
`modified_by` int(11) unsigned NOT NULL DEFAULT '0',
`checked_out` int(11) NOT NULL DEFAULT '0',
`checked_out_time` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
PRIMARY KEY (`id`),
KEY `idx_checkout` (`checked_out`),
KEY `idx_published` (`published`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Лістинг 4.3. DDL-код таблиці, де зберігаються дані послуг

Вигляд створеної таблиці із внесеними даними зображений на рисунку

4.11

id	department_id	category_id	hall_id	code	name	description	description_period	description_result	description_documents	description_files	duration_time	derzhposlугy	published
1	26	1	1	A-14.1-02	Надання містобудівних умов та обмежень забудови зе...		<p>Протягом семи робочих днів з дня реєстрації зая...	<p>Містобудівні умови та обмеження забудови земель...	<p>1. Заява</p><p>2. Засвідчена в установленому ...	<p align="center" style="text-align: justify;"><a ...	0		1
2	25	3	1	A-15-04	Надання дозволу на видалення зелених насаджень		50 днів	<div>У випадку – акт обстеження зелених насаджень...	<p style="text-align: justify;">У випадку видап...	<p align="center" style="text-align: left;"><a href...	0		1
3	25	3	1	A-15-05	Надання дозволу на перепоховання		30 (45) днів	<p>Рішення виконавчого комітету міської ради про н...	<p>1. Заява особи, яка здійснила поховання (корист...	<p align="center" style="text-align: justify;"><a ...	0		1
4	30	11	1	A-21-01	Видача посвідчень батьків та дитини з багатодітної...		<p>Протягом 10-ти днів після подання докумен...		<div style="text-align: justify;">Для вида...	<p align="center" style="text-align: left;"><a href...	0		1

Рисунок 4.4. Таблиця «Services»

DDL-код таблиці, де зберігаються категорії послуг, представлено у лістингу 4.2. Серед основних атрибутів потрібно відзначити наступні: `name` – назва категорії, `created` – дата створення, `modified` – дата модифікації.

DDL-код таблиці категорій послуг:

```

CREATE TABLE IF NOT EXISTS `cnap_office_services_categories` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `name` char(250) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `image` char(100) NOT NULL,
  `published` tinyint(1) NOT NULL DEFAULT '0',
  `ordering` int(11) NOT NULL DEFAULT '0',
  `created` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `created_by` int(11) unsigned NOT NULL DEFAULT '0',
  `created_by_alias` char(100) NOT NULL DEFAULT '',
  `modified` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `modified_by` int(11) unsigned NOT NULL DEFAULT '0',
  `checked_out` int(11) NOT NULL DEFAULT '0',
  `checked_out_time` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',

```

```
PRIMARY KEY (`id`),
KEY `idx_checkout` (`checked_out`),
KEY `idx_published` (`published`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Лістинг 4.4. DDL-код таблиці, де зберігаються категорії послуг.

Створена таблиця із внесеними даними зображена на рис. 4.5.

id	name	image	published	ordering	created	created_by	created_by_alias	modified	modified_by	checked_out	checked_out_time
1	Містобудування, архітектура та охорона культурної ...	images/office/categories/1.png	1	0	2015-09-26 10:47:40	996		2015-10-06 21:59:54	996	0	0000-00-00 00:00:00
2	Земельні питання	images/office/categories/2.png	1	0	2015-09-26 10:47:40	996		2015-10-10 12:11:53	996	0	0000-00-00 00:00:00
3	Житлово-комунальне господарство та екологія	images/office/categories/3.png	1	0	2015-09-26 10:47:40	996		2015-10-06 21:59:54	0	0	0000-00-00 00:00:00

Рисунок 4.5. Таблиця «Services Categories»

DDL-код таблиці, де зберігаються дані відносно календарних та вихідних днів, представлено у лістингу 4.5. Серед основних атрибутів потрібно відзначити наступні: `type` – тип (вихідний або робочий день), `created` – дата створення, `modified` – дата модифікації.

DDL-код таблиці вихідних днів:

```
CREATE TABLE IF NOT EXISTS `cnap_office_holidays` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `type` enum('recurrent','single','period') NOT NULL DEFAULT 'recurrent',
  `date_reccurent` date NOT NULL,
  `date_single` date NOT NULL,
  `date_period_1` date NOT NULL,
  `date_period_2` date NOT NULL,
  `published` tinyint(1) NOT NULL DEFAULT '0',
  `ordering` int(11) NOT NULL DEFAULT '0',
  `created` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `created_by` int(11) unsigned NOT NULL DEFAULT '0',
  `created_by_alias` char(100) NOT NULL DEFAULT '',
  `modified` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `modified_by` int(11) unsigned NOT NULL DEFAULT '0',
  `checked_out` int(11) NOT NULL DEFAULT '0',
  `checked_out_time` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  PRIMARY KEY (`id`),
  KEY `idx_checkout` (`checked_out`),
  KEY `idx_published` (`published`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Лістинг 4.5. DDL-код таблиці, де зберігаються категорії послуг

Створена таблиця із внесеними даними зображена на рисунку 4.6.

id	name	student_id	studyYear	semester	disciplines	published	ordering	created	created_by	created_by_alias	modified	modified_by
1	20140505-124434	139	2013	2	a:11:f:5;i:2;i:109;i:0;i:157;i:8;i:163;i:6;i:201;...	1	0	2014-05-05 08:44:34	1379		0000-00-00 00:00:00	0
2	20140505-124632	299	2013	2	a:8:f:14;i:3;i:76;i:1;i:109;i:0;i:157;i:3;i:180;i:...	1	0	2014-05-05 08:46:32	1379		0000-00-00 00:00:00	0
3	20140514-105100	74	2013	2	a:11:f:5;i:0;i:109;i:0;i:155;i:0;i:157;i:0;i:167;...	1	0	2014-05-14 06:51:00	1379		0000-00-00 00:00:00	0
4	20140516-134418	119	2013	2	a:11:f:5;i:0;i:109;i:0;i:157;i:0;i:163;i:0;i:201;...	1	0	2014-05-16 09:44:18	1379		0000-00-00 00:00:00	0
5	20140523-162139	216	2013	2	a:8:f:5;i:5;i:109;i:0;i:185;i:2;i:186;i:2;i:192;i:...	1	0	2014-05-23 12:21:39	1379		0000-00-00 00:00:00	0

Рисунок 4.6. Таблиця «Holidays»

Загальну структуру та зв'язки можна переглянути на діаграмі корпоративної моделі (рис. 4.7).

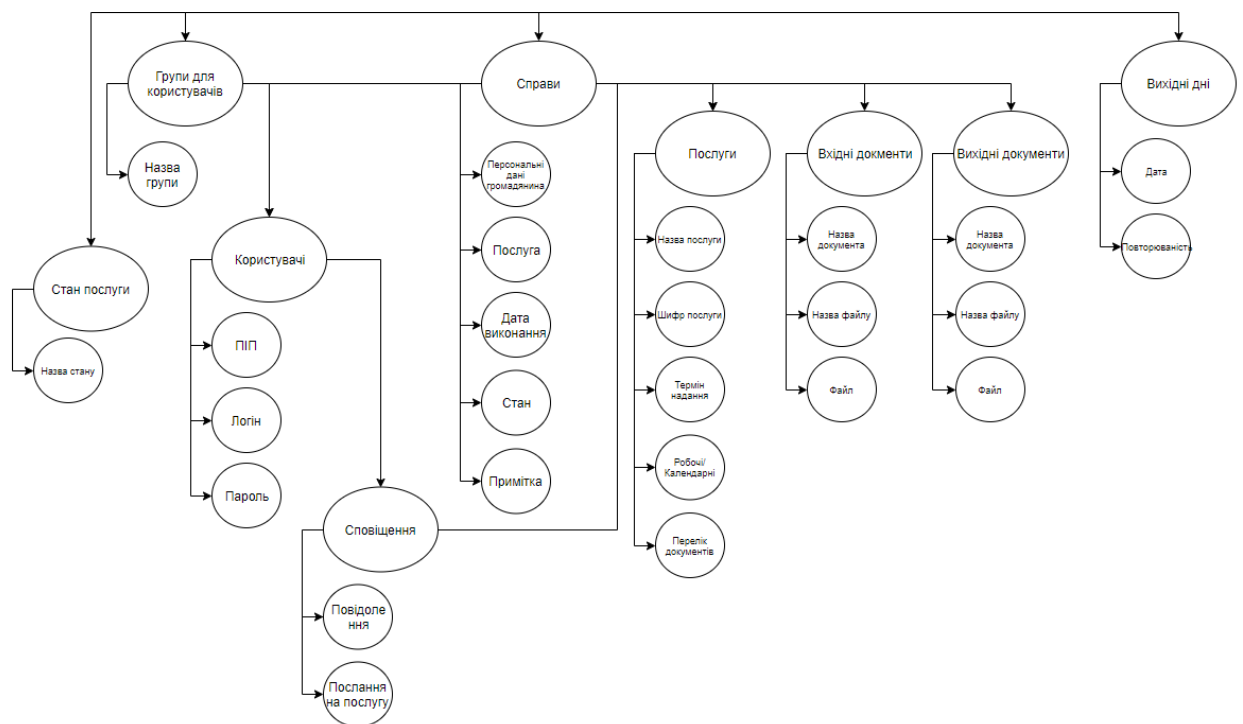


Рисунок 4.7. Діаграма редагування профілю користувача

Детальний перелік відношень та їх властивостей наведений в таблицях ідентифікаторів 4.1-4.9.

В таблиці 4.1 представлено опис атрибутів, їх тип даних та розмірність для сутності послуги.

Таблиця 4.1

Таблиця ідентифікаторів сутності «Послуги»

Властивості	Тип	Розмір	Ідентифікатор
Послуги			dov_services
Назва послуги	Текст	254	Nazva
Шифр послуги	Текст	30	Kod_posl
Термін надання	Число		termin
Тип днів в терміні надання	Логічний		Termin_t

В таблиці 4.2 представлено опис атрибутів, їх тип даних та розмірність для сутності «Групи для користувачів».

Таблиця 4.2

Таблиця ідентифікаторів сутності «Групи для користувачів»

Властивості	Тип	Розмір	Ідентифікатор
Групи для користувачів			dov_group
Назва групи	Текст	254	name

В таблиці 4.3. представлено опис атрибутів, їх тип даних та розмірність для сутності «Справи».

Таблиця 4.3

Таблиця ідентифікаторів сутності «Справи»

Властивості	Тип	Розмір	Ідентифікатор
Справи			cnap_reglist
Номер	Число		nn
Дата реєстрації	Дата		dat_pr_zaj
Дата виконання	Дата		dat_vuk
Тип особи	Текст	32	tup_osobu
Назва особи	Текст	254	nazva_osobu
Індиф. Код.	Текст	10	id_kod
Адреса	Текст	254	adr_osobu
Піб керівника	Текст	40	pib_kerivnuka
Телефон	Текст	15	tel
Факс	Текст	15	fax
Мобільний	Текст	15	mob
Електронна пошта	Текст	40	e1mail
Назва об'єкта	Текст	254	nazva_obj
Адреса об'єкта	Текст	254	adr_obj
Стан послуги	Текст	30	stan_doc
Дата зміни стану	Дата		dat_stan
Результат	Текст	30	nn_stan
Дата повернення документа	Дата		dat_pov_doc
Дата видачі документа	Дата		dat_vud_zaj
Примітка	Текст	254	prumitka

В таблиці 4.4. представлено опис атрибутів, їх тип даних та розмірність для сутності «Стан послуги».

Таблиця 4.4

Таблиця ідентифікаторів сутності «Стан послуги»

Властивості	Тип	Розмір	Ідентифікатор
Стан послуги			dov_status
Назва статусу	Текст	30	name

В таблиці 4.5 представлено опис атрибутів, їх тип даних та розмірність для сутності «Стан послуги».

Таблиця 4.5

Таблиця ідентифікаторів сутності «Користувачі»

Властивості	Тип	Розмір	Ідентифікатор
Користувачі			dov_users
Логін	Текст	40	login
Пароль	Текст	40	pass
ППП	Текст	60	vukonavec
Тип користувача	Логічний		user_type

В таблиці 4.6. представлено опис атрибутів, їх тип даних та розмірність для сутності «Стан послуги».

Таблиця 4.6

Таблиця ідентифікаторів сутності «Сповіщення користувачів»

Властивості	Тип	Розмір	Ідентифікатор
Сповіщення користувачів			dov_alerts
Дата створення сповіщення	Дата		date
Повідомлення сповіщення	Текст	254	message

В таблиці 4.7 представлено опис атрибутів, їх тип даних та розмірність для сутності «Вихідні дні»

Таблиця 4.7

Таблиця ідентифікаторів сутності «Вихідні дні»

Властивості	Тип	Розмір	Ідентифікатор
Вихідні дні			dov_wknd
Вихідний день	Дата		date
Повторюваність	Логічний		repeats

В таблиці 4.8 представлено опис атрибутів, їх тип даних та розмірність для сутності «Вхідні документи».

Таблиця 4.8

Таблиця ідентифікаторів сутності «Вхідні документи»

Властивості	Тип	Розмір	Ідентифікатор
Вхідні документи			indoc
Назва документа	Текст	254	name
Назва файлу	Текст	254	docname
Файл	Байт		file

В таблиці 4.9 представлено опис атрибутів, їх тип даних та розмірність для сутності «Вихідні документи».

Таблиця 4.9

Таблиця ідентифікаторів сутності «Вихідні документи»

Властивості	Тип	Розмір	Ідентифікатор
Вихідні документи			outdoc
Назва документа	Текст	254	name
Назва файлу	Текст	254	docname
Файл	Байт		file

Загальну ER-діаграму представлено на рисунку 4.8, яка візуально представляє зв'язок між атрибутами у відношеннях.

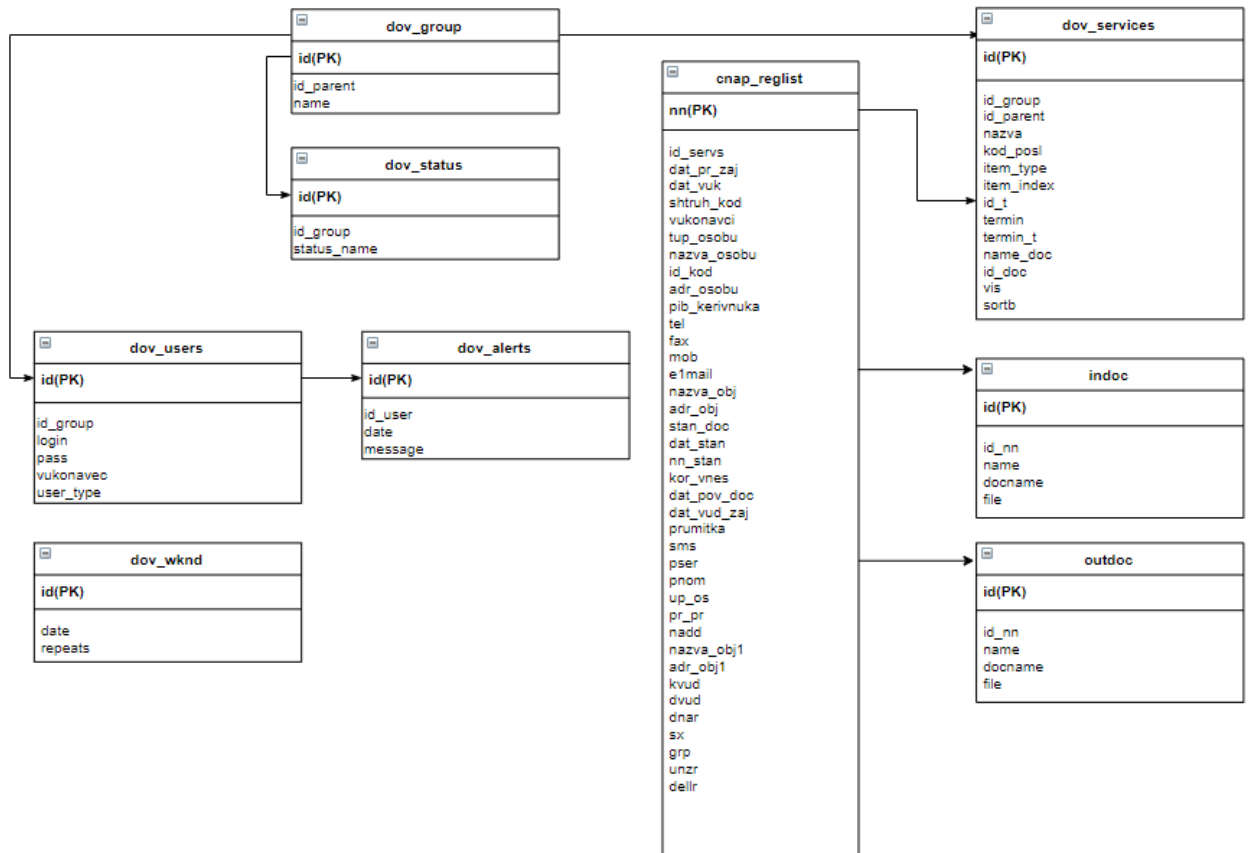


Рисунок 4.8. ER-діаграма

Для бля більшої швидкості при вибірці використовуються різні види індексів (повнотекстові, звичайні, унікальні). Індеси необхідні для того, щоб можна було швидко і зручно звертатись до певного виділеного масиву даних. Також не потрібно зловживати і додавати зайвих індексів, адже це може вповільнити швидкість стандартних операцій із запитам (додавання, оновлення) за рахунок ще одного поля.

Поля відношень, наведених нижче, використано в якості індексів, фізична реалізація яких подана у пункті «Фізичне проектування». Індеси

були створені по тих полях, по яких найчастіше проходить фільтрація і вибірка даних. Список таких індексів наведена нижче:

- snap_reglist.nn
- snap_reglist.id_serve
- dov_services.id
- dov_services.id_group
- dov_services.id_parent
- indoc.id
- indoc.id_nn
- outdoc.id
- outdoc.id_nn
- dov_alerts.id
- dov_alerts.id_user

DDL скрипти для генерування реляційних відношень і ключів поданий в додатку Б.

Оскільки система повинна надавати віддалений доступ до інформації та спеціальних можливостей, вона розробляється у вигляді веб-сайту. Також в процесі розробки приділяється увага не лише доступу з домашніх комп'ютерів, але і з мобільних пристроїв. Це дасть можливість зручно переглядати інформацію, реєструватись у чергу, перевіряти стан виконання послуг, наприклад, у дорозі з мобільного телефона.

Для функціонування системи потрібна PHP 7.2, база даних MySQL 5.5. В налаштуваннях PHP має бути увімкнений модуль Mbstring та XMLReader, функція Iconv, вимкнені Magic Quotes та Register Globals. Система розміщена на сервері під управлінням Apache 2.4.10. Мінімальні вимоги до апаратного забезпечення: оперативна пам'ять – 4Гб, 1Гб дискового простору.

Програмна система побудована з використанням MVC-підходу проектування. На рисунку 4.9 зображено перелік класів моделі. Окрім цього є

ще класи первинного і вторинного контролерів, таблиць, специфічних полів форм, а також бібліотек функцій.

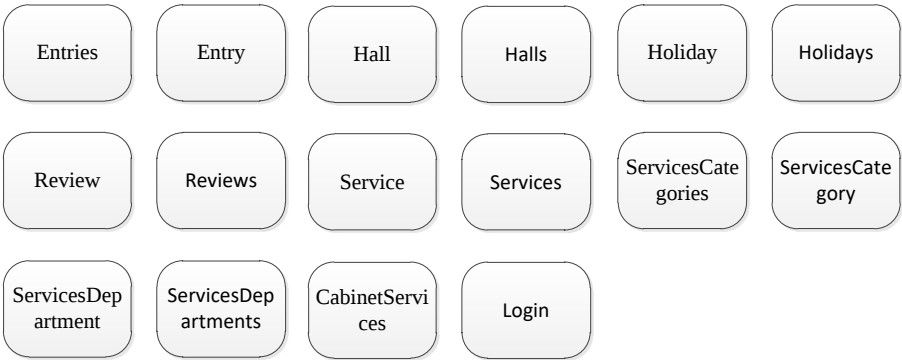


Рисунок 4.9. Перелік класів

На рисунку 4.10 наведена діаграма основних класів для реалізації бізнес-логіки.

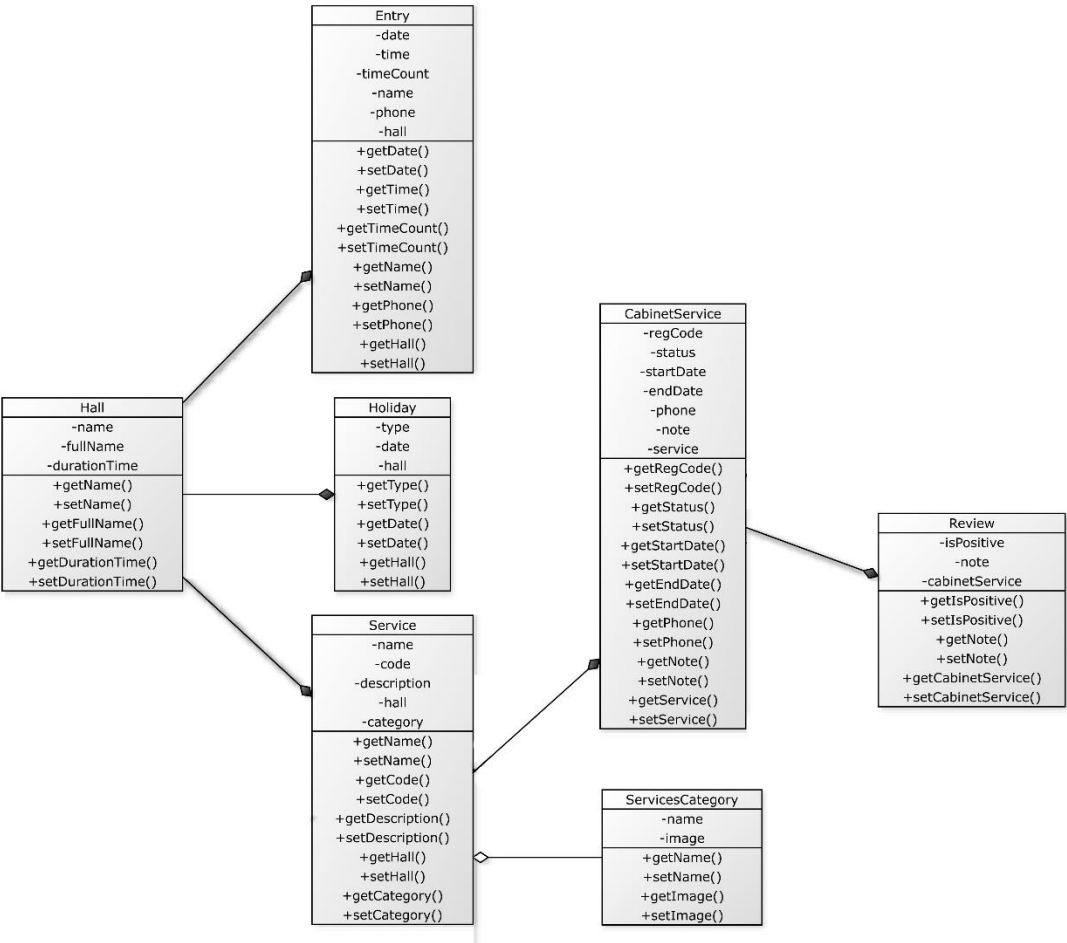


Рисунок 4.10. Діаграма класів

Далі наведений код моделі даних для роботи із створенням та збереженням інформації послуги.

```

class OfficeModelHoliday extends JModelAdmin{
public function getTable($type='Holidays', $prefix='OfficeTable',
$config=array()) {
    return JTable::getInstance($type, $prefix, $config);
}

public function getForm($data=array(), $loadData=true) {
    $form=$this->loadForm('com_office.holiday', 'holiday',
array('control'=>'jform', 'load_data'=>$loadData));
    if(empty($form)) {
        return false;
    }
    return $form;
}

protected function loadFormData()
{
    $data=JFactory::getApplication()-
>getUserState('com_office.edit.holiday.data', array());
    if(empty($data)) {
        $data=$this->getItem();
    }

    // Date Reccurent.
    @list($month, $day) = explode('-', $data->date_reccurent, 3);
    $data->date_reccurent = array(
        'month' => !empty($month)? $month : 1,
        'day'    => !empty($day)? $day : 1,
    );

    // Date Single.
    if ($data->date_single == '0000-00-00') {
        $data->date_single = '';
    }

    // Date Period.
    if ($data->date_period_1 == '0000-00-00') {
        $data->date_period_1 = '';
    }
    if ($data->date_period_2 == '0000-00-00') {
        $data->date_period_2 = '';
    }

    return $data;
}

protected function prepareTable(&$table)
{
    // Date Reccurent.

```

```

        $table->date_reccurent =
!empty(JRequest::getVar('jform')['date_reccurent'])? $table-
>date_reccurent : null;
        if(is_array($table->date_reccurent)) {
            $table->date_reccurent = date('Y')
                . '-' . (!empty($table->date_reccurent['month'])?
$хtable->date_reccurent['month'] : 1)
                . '-' . (!empty($table->date_reccurent['day'])? $table-
>date_reccurent['day'] : 1);
        }
    }
}

```

Цей код лаконічний за рахунок того, що унаслідкується від добре пропрацьованої моделі JModelAdmin, а структуру форми і властивостей кожного поля бере з відповідного xml-файла (holiday.xml):

```

<?xml version="1.0" encoding="utf-8"?>
<form>
<fieldset
addfieldpath="/administrator/components/com_office/models/fields"
>

<field name="id" type="hidden" default="0" class="readonly"
    readonly="true" />

<field name="type" type="list" default="recurrent"
    size="1"
    label="COM_OFFICE_TYPE" description="COM_OFFICE_TYPE"
    required="true" >

    <option value="recurrent">COM_OFFICE_TYPE_RECURRENT</option>
    <option value="single">COM_OFFICE_TYPE_SINGLE</option>
    <option value="period">COM_OFFICE_TYPE_PERIOD</option>
</field>

<field name="date_reccurent" type="officereccurentdate"
    label="COM_OFFICE_DATE" description="COM_OFFICE_DATE" />

<field name="date_single" type="calendar"
    format="%Y-%m-%d"
    label="COM_OFFICE_DATE" description="COM_OFFICE_DATE" />

<field name="date_period_1" type="calendar"
    format="%Y-%m-%d"
    label="COM_OFFICE_DATE" description="COM_OFFICE_DATE" />

<field name="date_period_2" type="calendar"
    format="%Y-%m-%d"
    label="COM_OFFICE_DATE" description="COM_OFFICE_DATE" />

<field name="published" type="list" default="1"
    size="1"

```

```

class="chzn-color-state"
label="JSTATUS" description="JSTATUS">

<option value="1">JPUBLISHED</option>
<option value="0">JUNPUBLISHED</option>
<option value="-2">JTRASHED</option>
</field>

```

UML-діаграму класів здійснення основних операцій з інформаційними структурами, додавання, оновлення та видалення інформації з бази даних представлено на рисунку 4.11.

Розроблена система може бути легко інтегрована в існуючі інформаційні системи центру надання адміністративних послуг, оскільки реалізована із застосуванням сучасних інформаційних технологій, а реалізовані програмні компоненти легко інтегруються та масштабуються.

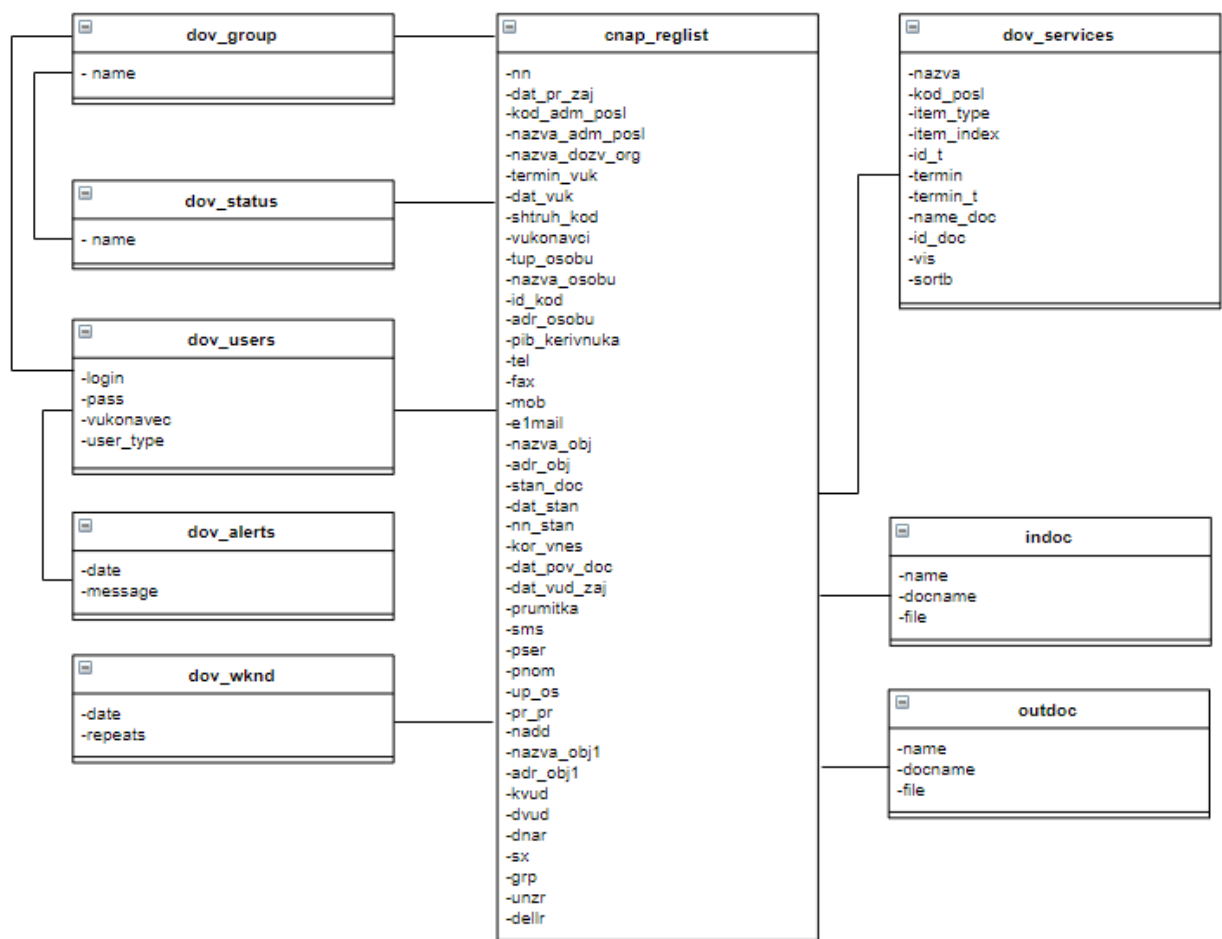


Рисунок 4.11.UML-діаграму класів

Реалізовані рішення поряд з підсистемою інтелектуального аналізу даних сформували цілісний програмно-аналітичний комплекс.

4.3. Особливості організації користувацького інтерфейсу

Дизайн сайту виконаний відповідно до корпоративних правил (верстка, логотип, кольорова гамма). Головна сторінка сайту зображена на рисунку 4.12.

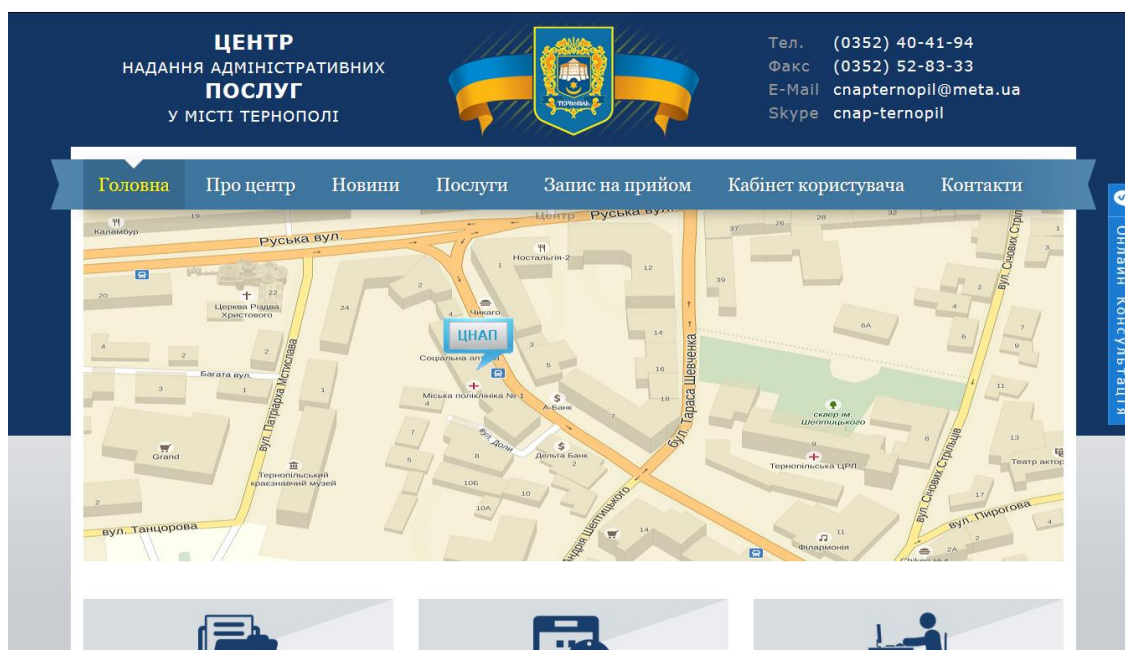


Рисунок 4.12. Головна сторінка сайту

Довідник послуг є одним із найбільш корисних і спеціалізованих розділів сайту, у якому користувачі зможуть знайти основну інформацію, яка їм знадобиться при візиті у ЦНАП. Оскільки послуг багато і відвідувач може погано орієнтуватись у них, то для більш комфортного пошуку інформація розбита на три основні сторінки: послуги за категоріями, послуги за суб'єктами надання (установами) та реєстр усіх послуг. Навігація по

сторінках відбувається дуже швидко за рахунок AJAX-запитів. Вигляд сторінки із категоріями зображений на рисунку 4.13.

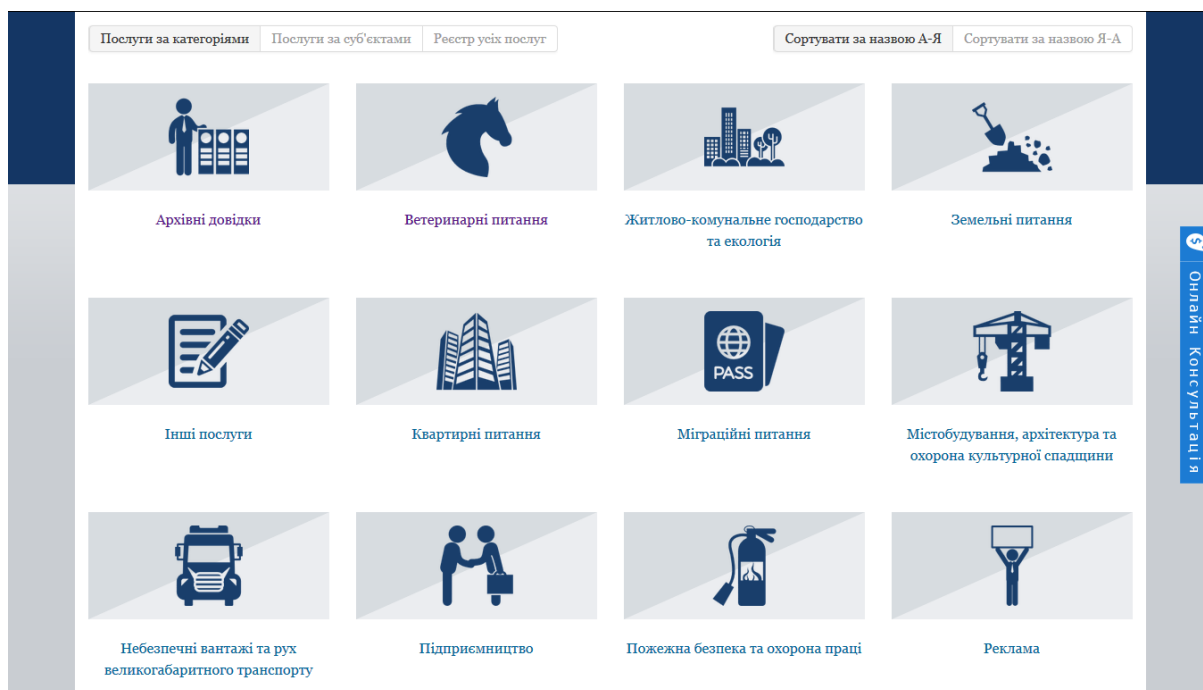


Рисунок 4.13. Послуги за категоріями

На рисунку 4.14 зображений список усіх послуг, де окрім стандартного сортування за назвою також є сортування за залами і швидкий пошук по назві чи шифру послуги.

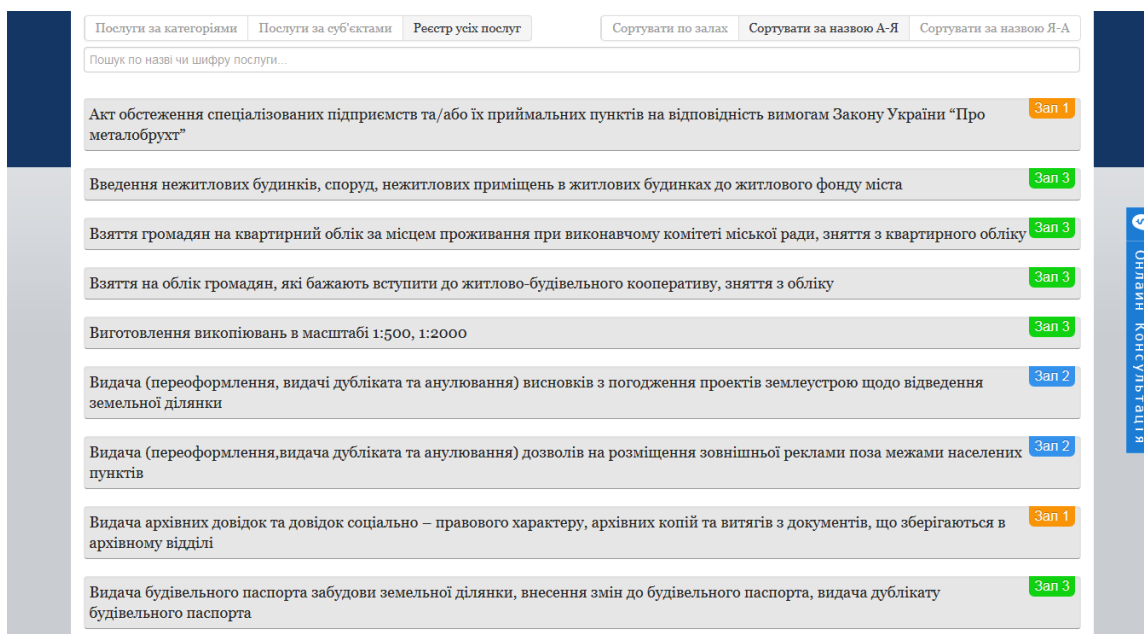


Рисунок 4.14. Реєстр усіх послуг

При переході на сторінку послуги (рис. 4.15) можна переглянути детальну інформацію, скачати пов'язані файли, роздрукувати, перейти до даної послуги на офіційний державний сайт. Також тут приведений графік запису в електронну чергу (який наразі відрізняється від повного робочого графіку) і одразу можна перейти до реєстрації. Якщо відвідувач має заповнені документи, він може відправити їх електронні версії на перевірку, скориставшись пунктом «Перевірка документів онлайн».

Вичерпний перелік документів

1. Заява;
2. Перелік харчових продуктів , що плануються виробляти чи переробляти;
3. Квитанція про оплату.

Файли (інформація про послугу, інформаційна картка, зразки заяв)

ІНФОРМАЦІЙНА КАРТКА з видачі експлуатаційного дозволу оператором ринку, що проводять діяльність, пов'язану з виробництвом та переробкою харчових продуктів тваринного походження

ФОРМА ЗАЯВИ ДО ПОСЛУГИ ДЛХ 1.4/15-1

➔ Відкрити для перегляду

Перевірка документів онлайн Запис на прийом

* Попередній запис на вівторок та п'ятницю не передбачений. Щоденний графік роботи центру можна переглянути на сторінці [Графік роботи](#)

МИ НАДАЄМО ПОСЛУГИ ЯКІСНО ТА ШВИДКО

★★★★★

Онлайн Консультація

Рисунок 4.16. Сторінка послуги

На сторінці реєстрації в електронній черзі (рис. 4.17) потрібно обрати дату і одну з вільних годин. Список днів формується на основі графіка запису в електронну чергу, який задається для кожного залу індивідуально. Також на цей список впливають вихідні та святкові дні. Для кожного залу задається тривалість надання однієї послуги на основі якої години розбиваються на рівні проміжки часу (наразі 30хв.).

Головна Про центр Новини Послуги **Запис на прийом** Кабінет користувача Контакти

Ви тут: Головна сторінка > Вибір залу > Вибір дати і години

Зал 2 - Сектор дозвільних процедур

10.02.2016 (Середа)							
09:00	09:30	10:00	10:30	11:00	11:30	12:00	12:30

11.02.2016 (Четвер)					
14:00	14:30	15:00	15:30	16:00	16:30

15.02.2016 (Понеділок)							
09:00	09:30	10:00	10:30	11:00	11:30	12:00	12:30

17.02.2016 (Середа)							
09:00	09:30	10:00	10:30	11:00	11:30	12:00	12:30

18.02.2016 (Четвер)					
14:00	14:30	15:00	15:30	16:00	16:30

* Виберіть одну із вільних годин

Онлайн Консультація

Рисунок 4.17. Вибір дати і години

На рисунку 4.18 зображений наступний після вибору дати і години етап реєстрації в електронній черзі – заповнення реєстраційної форми. Тут потрібно ввести свої персональні дані за якими можна буде ідентифікувати клієнта. Якщо користувач авторизований на сайті, то ПІБ і номер телефону із його профілю автоматично будуть підставлені в ці поля. У випадковому списку можна вказати кількість питань для вирішення. Це дасть можливість оператору центру краще орієнтуватися в тому, скільки потрібно часу на обслуговування даного клієнта. Також користувачу потрібно дати згоду на збір та обробку його даних, які, до речі, автоматично видаляються із сайту через декілька днів після проходження дати. Якщо усе заповнено вірно, на наступній сторінці буде повідомлення про успішну реєстрацію та можливість роздрукувати реєстраційний чек.

Головна Про центр Новини Послуги **Запис на прийом** Кабінет користувача Контакти

Ви тут: Головна сторінка > Вибір залу > Вибір дати і години > Форма запису

Реєстрація на прийом

Обраний зал	Зал 2 - Сектор дозвільних процедур
Обраний день	10.02.2016 (Середа)
Обрана година	10:00

Форма запису

Введіть своє прізвище та ім'я

Введіть свій номер мобільного телефону

Оберіть кількість питань для вирішення

☐ Даю згоду на збір та обробку моїх персональних даних для надання послуги реєстрації в електронній черзі

Зареєструватися

* Для реєстрації в електронній черзі потрібно заповнити форму

Рисунок 4.19. Реєстрація на прийом

На рисунку 4.19 зображена форма реєстрації, яку потрібно пройти, щоб отримати доступ до всіх функцій кабінету користувача. Логін – це індивідуальне слово для входу в кабінет. Це може бути набір символів, номер телефону, емейл чи щось інше, що буде зручно запам'ятати користувачу. Адреса електронної пошти є необов'язковим полем. Якщо вона введена, то з її допомогою можна буде отримати повідомлення від ЦНАПу, використати для відновлення забутого логіна чи пароля.

Реєстрація користувача

ПІБ

Логін

Пароль

Підтвердити пароль

Адреса електронної пошти (за бажанням)

Зареєструватися **Скасувати**

(Якщо після натиснення кнопки "Зареєструватися" нічого не відбулося, перевірте введені дані, поля з помилкою відзначились червоним кольором.)

Рисунок 4.20. Форма реєстрації в кабінеті користувача

Після реєстрації чи авторизації на сайті відвідувач потрапляє на сторінку із переліком доступних для нього функцій (рис. 4.21).

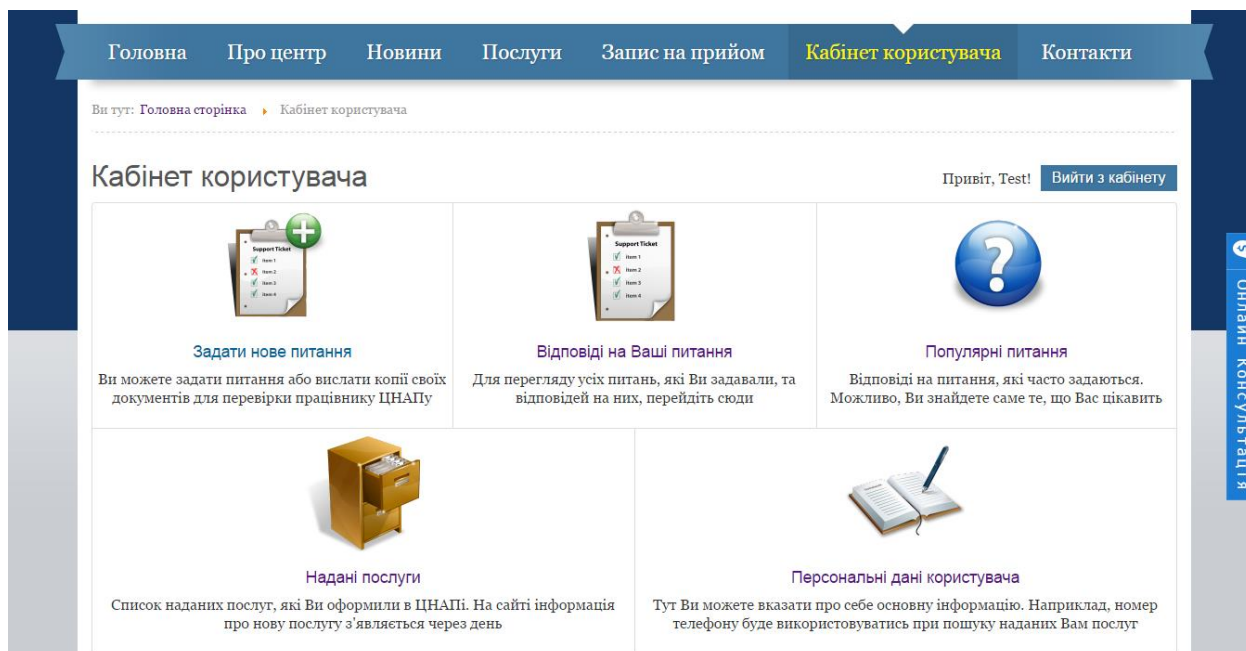


Рисунок 4.21. Головне меню кабінету користувача

«Задати нове питання» відкриває для користувача форму створення нової заявки в службу підтримки (рис. 4.22). Тут можна обрати категорію послуги, до якої стосується запитання, вказати тему та текст повідомлення, прикріпити електронні варіанти документів. Якщо користувач перейшов на цю форму не з меню кабінету, а з пункту «Перевірка документів онлайн» на сторінці послуги, для зручності всі поля будуть автоматично заповнені, потрібно буде лише прикріпити електронні документи.

Ви тут: [Головна сторінка](#) > [Кабінет користувача](#) > [Задати нове питання](#)

Нова заявка в службу підтримки

[Задати нове питання](#) [Відповіді на Ваші питання \(7\)](#)

Деталі повідомлення

Категорія:

Тема:

Текст Вашого повідомлення

Відправляю документи на перевірку до послуги:
"Видача архівних довідок та довідок соціально – правового характеру, архівних копій та витягів з документів, що зберігаються в архівному відділі"

Ваші документи [Завантажити файли](#) [Перетягніть сюди файли](#)

[Відправити заявку](#)

Рисунок 4.22. Форма нової заявки

На сторінці «Відповіді на Ваші питання» можна переглянути список усіх заявок, які задавав користувач в службу підтримки, їх статуси та час останньої активності (рис. 4.23).

Ви тут: [Головна сторінка](#) > [Кабінет користувача](#) > [Задати нове питання](#)

Ваші заявки в службу підтримки

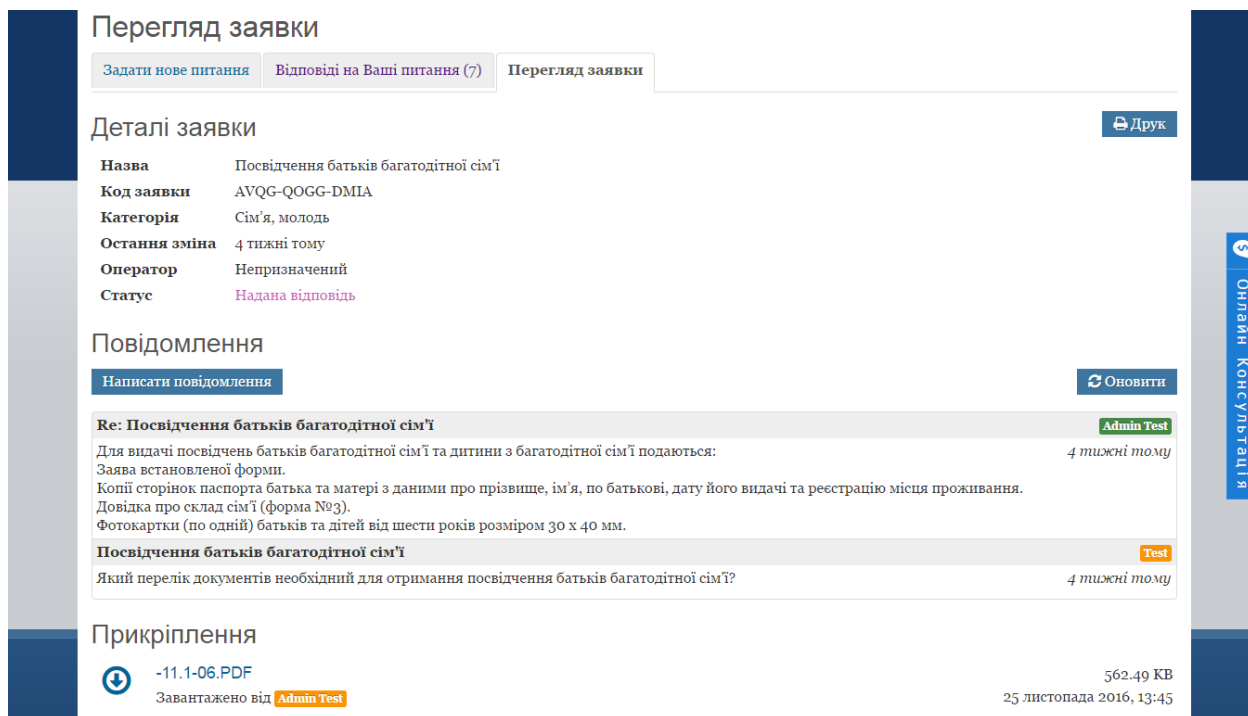
[Задати нове питання](#) [Відповіді на Ваші питання \(7\)](#)

Назва	Категорія	Статус	Остання зміна
Зал 1. Питання до послуги: Н-26-02	Архівні довідки	Відкрито	1 день тому
Зал 1. Питання до послуги: Н-26-02	Архівні довідки	Відкрито	1 день тому
Посвідчення батьків багатодітної сім'ї	Сім'я, молодь	Надана відповідь	4 тижні тому
Земельне питання 2	Земельні питання	Надана відповідь	4 тижні тому
Зал 1. Питання до послуги: Н-26-02	Архівні довідки	Відкрито	4 тижні тому
Тестове питання 2	Питання без категорії	Відкрито	4 тижні тому
Тестове питання	Земельні питання	Відкрито	4 тижні тому

Показати:

Рисунок 4.23. Список усіх питань

На рис. 4.24 показана сторінка однієї заявки. Тут ведеться переписка між користувачем та оператором, є можливість прикріпляти додаткові файли, переглядати більш детальну інформацію щодо заявки.



Перегляд заявки

Задати нове питання | **Відповіді на Ваші питання (7)** | Перегляд заявки

Деталі заявки Друк

Назва	Посвідчення батьків багатодітної сім'ї
Код заявки	AVQG-QOGG-DMIA
Категорія	Сім'я, молодь
Остання зміна	4 тижні тому
Оператор	Непризначений
Статус	Надана відповідь

Повідомлення Оновити

[Написати повідомлення](#)

Re: Посвідчення батьків багатодітної сім'ї Admin Test

Для видачі посвідчень батьків багатодітної сім'ї та дитини з багатодітної сім'ї подаються:
Заява встановленої форми.
Копії сторінок паспорта батька та матері з даними про прізвище, ім'я, по батькові, дату його видачі та реєстрацію місця проживання.
Довідка про склад сім'ї (форма №3).
Фотокартки (по одній) батьків та дітей від шести років розміром 30 x 40 мм.

4 тижні тому

Посвідчення батьків багатодітної сім'ї Test

Який перелік документів необхідний для отримання посвідчення батьків багатодітної сім'ї?

4 тижні тому

Прикріплення

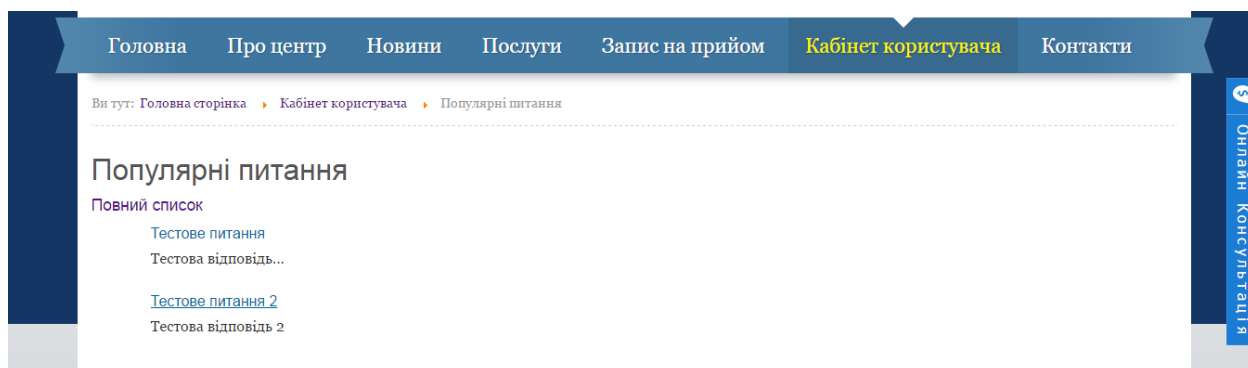
 -11.1-06.PDF Admin Test

Завантажено від 562.49 KB
25 листопада 2016, 13:45

Онлайн Консультація

Рисунок 4.24. Перегляд заявки

Перед тим, як задавати власне питання, користувач має змогу переглянути список питань, які задаються дуже часто (рис. 4.25).



Головна | Про центр | Новини | Послуги | Запис на прийом | **Кабінет користувача** | Контакти

Ви тут: Головна сторінка > Кабінет користувача > Популярні питання

Популярні питання

[Повний список](#)

[Тестове питання](#)
Тестова відповідь...

[Тестове питання 2](#)
Тестова відповідь 2

Онлайн Консультація

Рисунок 4.25. Популярні питання

Клієнти, які прийшли на прийом і отримали певні послуги, можуть в будь-який час зайти на сайті в список наданих послуг, ввести номер телефону, який вони лишили при отриманні послуги, та реєстраційний номер з опису документів, перевірити стан виконання послуг. Тобто тепер немає потреби телефонувати чи приходити у ЦНАП лише для того, щоб дізнатись стан виконання, що значно економить час. До слова, інколи послуги надаються досить довго (місяць і більше). На рисунку 4.26 зображений список послуг користувача, їх основна інформація та стан.

34789	Надання висновку про погодження проекту землеустрою щодо відведення земельної ділянки (Н-14.2-13)	Дата прийняття: 27-01-2016 Термін виконання: 10 Дата виконання: 09-02-2016	Задоволено	Лишити відгук ▾
34790	Оформлення паспорта прив'язки тимчасової споруди, продовження строку дії паспорта прив'язки (Н-14.2-08)	Дата прийняття: 27-01-2016 Термін виконання: 10 Дата виконання: 09-02-2016	Відмовлено	Лишити відгук ▾
34791	Оформлення паспорта прив'язки тимчасової споруди, продовження строку дії паспорта прив'язки (Н-14.2-08)	Дата прийняття: 27-01-2016 Термін виконання: 10 Дата виконання: 09-02-2016	Задоволено	Лишити відгук ▾
34792	Оформлення паспорта прив'язки тимчасової споруди, продовження строку дії паспорта прив'язки (Н-14.2-08)	Дата прийняття: 27-01-2016 Термін виконання: 10 Дата виконання: 09-02-2016	Відмовлено	Лишити відгук ▾
34793	Державна реєстрація земельної ділянки з видачею витягу з Державного земельного кадастру (АДГК 20-01)	Дата прийняття: 26-01-2016 Термін виконання: 14 к/днів Дата виконання: 09-02-2016		Лишити відгук ▾
34794	Наказ про надання земельної ділянки (у власність, користування, в тому числі на умовах оренди) та затвердження документації із землеустрою, який відповідно до статті 122 Земельного кодексу України видається Головним управлінням Леджтеокалестов /ЛЛХ	Дата прийняття: 26-01-2016 Термін виконання: 14	Задоволено	Лишити відгук ▾

Рисунок 4.26. Надані послуги

Також клієнт має можливість залишити позитивний чи негативний відгук. Для того створено відповідні умови, процес займе мінімум часу та є максимально простим.

На сторінці «Персональні дані користувача» (рис. 4.27) можна вказати про себе основну інформацію. Наприклад, номер телефону буде використовуватись при пошуку наданих послуг чи реєстрації на прийом.

Рисунок 4.27. Сторінка редагування персональної інформації

Кабінет користувача для адміністратора має додатковий пункт «Адміністрування». Також зверху сторінки відображається коротка інформація про питання, які потрібно опрацювати (рис. 4.28).

 Адміністрування Можливість відповідати на питання, редагувати матеріали	 Задати нове питання Ви можете задати питання або вислати копії своїх документів для перевірки працівнику ЦНАПу	 Відповіді на Ваші питання Для перегляду усіх питань, які Ви задавали, та відповідей на них, перейдіть сюди
 Популярні питання Відповіді на питання, які часто задаються. Можливо, Ви знайдете саме те, що Вас цікавить	 Надані послуги Список наданих послуг, які Ви оформили в ЦНАПі. На сайті інформація про нову послугу з'являється через день	 Персональні дані користувача Тут Ви можете вказати про себе основну інформацію. Наприклад, номер телефону буде використовуватись при пошуку наданих Вам послуг

Рисунок 4.28. Кабінет користувача для адміністратора

На рисунку 4.29 показана сторінка зі списком питань від користувачів. В системі є можливість давати права для адміністратора бачити всі питання, або лише ті, які були призначені саме для нього.

Ви тут: [Головна сторінка](#) > [Кабінет користувача](#) > [Підтримка](#)

Ваші заявки в службу підтримки

[Огляд](#) [Підтримка](#) [Зміст](#)

Відкрито (5) Надаана відповідь (2) В процесі (0) Інші ▾

Створити заявку для **Зареєстрований користувач** Інструменти ▾

Пошук заявок Знайти Відмітини - Фільтр - ▾

Теги заявок Теги ▾ Теги не вибрані

Код заявки	Статус	Користувач	Остання зміна	
Зал 1. Питання до послуги: Н-26-02 DOGV-SMJD-NBQO	Відкрито	Test	1 день тому	Непризначений 1
Зал 1. Питання до послуги: Н-26-02 QZYD-QRJW-RZCT	Відкрито	Test	1 день тому	Непризначений 1
Зал 1. Питання до послуги: Н-26-02 UKYX-AZZQ-KEGD	Відкрито	Test	4 тижні тому	Непризначений 1
Тестове питання 2 ITXD-KPEK-RPQX	Відкрито	Test	4 тижні тому	Непризначений 1
Тестове питання GSKV-RIOJ-NXCU	Відкрито	Test	4 тижні тому	Непризначений 2

[Мої заявки](#) [Інші заявки учасника](#) [Непризначений](#)

Рисунок 4.29. Список питань від користувачів

Список популярних запитань адміністратор формує сам на свій розсуд. Це можна зробити на сторінці редагування частих запитань (рис. 4.30).

Ви тут: [Головна сторінка](#) > [Кабінет користувача](#)

Управління контентом

[Огляд](#) [Підтримка](#) [Зміст](#)

Часті питання

Часті питання

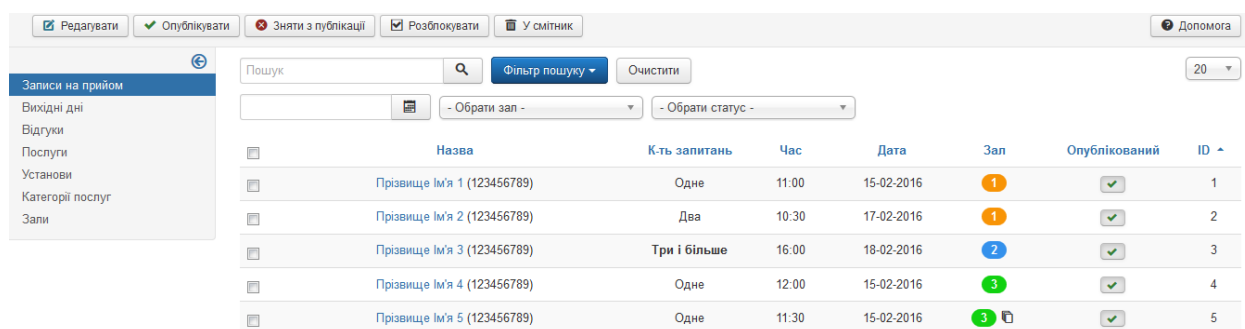
Вперед Відмітини Опублікувати? ▾ Виберіть корис: ▾ Категорія ▾ Вибір доступу ▾ Новий

Питання	Категорія	Featured	Access	Опублікувати	Автор
Тестове питання 2	Повний список	✗	Public	✓	Super User
Тестове питання	Повний список	✗	Public	✓	Super User

Рисунок 4.30. Редагування списку частих запитань

Для працівників центру зроблений окремий адміністративний інтерфейс із гнучким розмежуванням прав доступу.

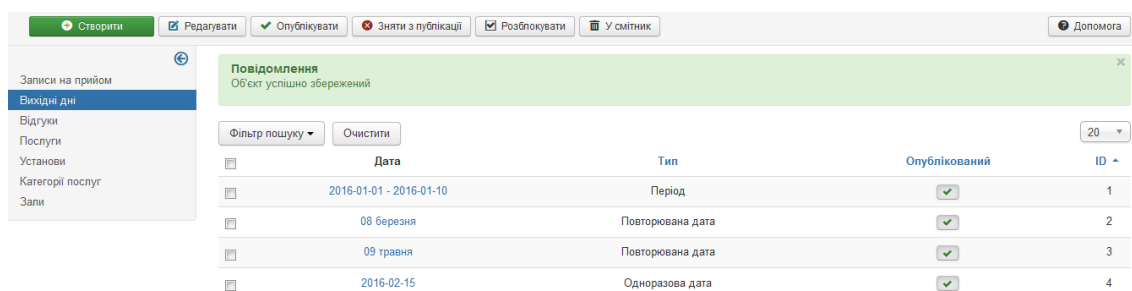
На даній сторінці можна переглянути останні або найбільш популярні статті, інших адміністраторів, які в цей час теж знаходяться онлайн. З цього інтерфейсу можна створювати новини, редагувати існуючі матеріали, додавати нових користувачів, створювати опитування. Спеціалізованим для ЦНАП розділом є «Послуги центру». На рис. 4.31 зображений список записів на прийом. Для того, щоб було зручно переглянути відвідувачів на певну годину та зал, можна скористатись фільтром зверху списку. Якщо видно, що дані запису невірні, його можна анулювати, знявши з публікації (нажати на зелену галочку), або просто видалити. Тоді вказана година знову стане доступною для реєстрації.



Назва	К-ть запитань	Час	Дата	Зал	Опублікований	ID
Прізвище Ім'я 1 (123456789)	Одне	11:00	15-02-2016	1	✓	1
Прізвище Ім'я 2 (123456789)	Два	10:30	17-02-2016	1	✓	2
Прізвище Ім'я 3 (123456789)	Три і більше	16:00	18-02-2016	2	✓	3
Прізвище Ім'я 4 (123456789)	Одне	12:00	15-02-2016	3	✓	4
Прізвище Ім'я 5 (123456789)	Одне	11:30	15-02-2016	3	✓	5

Рисунок 4.31. Записи на прийом

Для того, щоб заборонити реєстрацію в електронну чергу на певний день чи період, потрібно створити відповідний запис в списку вихідних днів (рис. 4.32).



Дата	Тип	Опублікований	ID
2016-01-01 - 2016-01-10	Період	✓	1
08 березня	Повторювана дата	✓	2
09 травня	Повторювана дата	✓	3
2016-02-15	Одноразова дата	✓	4

Рисунок 4.32. Вихідні дні

На рисунку 4.33 зображена сторінка з усіма відгуками, які відправили користувачі зі своїх кабінетів. Для зручності можна відфільтрувати список, щоб побачити, наприклад, лише негативні відгуки для певного залу.

Тип	Дата	Номер	Послуга	Зал	Опублікований	ID
Позитивний	11-02-2016	35009	Декларація про початок виконання будівельних робіт (АББК-04-01)	2	✓	6
Позитивний	11-02-2016	35057	Декларація про початок виконання будівельних робіт (АББК-04-01)	2	✓	5
Негативний Відмовлено у наданні послуги	11-02-2016	35086	Надання містобудівних умов та обмежень забудови земельної ділянки (А-14.1-02)	1	✓	4
Позитивний	11-02-2016	35235	Видача витягу з технічної документації про нормативну грошову оцінку земельної ділянки (АДГК 20-12)	2	✓	3
Позитивний	11-02-2016	35228	Погодження дозволу на розміщення зовнішньої реклами, погодження зовнішнього вигляду вивіски, таблички (Н-14.1-19)	3	✓	2
Позитивний	11-02-2016	35233	Видача витягу з технічної документації про нормативну грошову оцінку земельної ділянки (АДГК 20-12)	2	✓	1

Рисунок 4.33. Відгуки

Список послуг (рис. 4.34) дозволяє переглянути основну інформацію про кожну з них, додати нову чи відредагувати існуючі. Для зручності можна фільтрувати список за категоріями, установами, залами. Тут вказана кількість позитивних та негативних відгуків про кожну з послуг, що допоможе одразу бачити лідируючі щодо якості надання та проблемні послуги, які потребують покращення.

Назва	Установа	Категорія послуг	Зал	Лайки	Дислайки	Опублікований	ID
Укладання договору оренди землі (А-25-12)	Відділ земельних ресурсів	Земельні питання	Зал 1	0	0	✓	14
Укладання договору на користування стоянками таксі, що знаходяться в комунальній власності, для перевезення пасажирів та їх багажу на таксі в м. Тернополі (Н-12.1-02)	Управління транспорту, комунікацій та зв'язку	Інші послуги	Зал 3	0	0	✓	38
Спеціальний дозвіл – лісорубний квиток на заготовку деревини під час проведення рубок головного користування (ДДХ 1.4/21-1)	Тернопільське обласне управління лісового та мисливського господарства	Житлово-комунальне господарство та екологія	Зал 2	0	0	✓	86
Розгляд відповідності намірів щодо місця розташування тимчасової споруди (Н-14.2-07)	Управління містобудування, архітектури та кадастру	Містобудування, архітектура та охорона культурної спадщини	Зал 3	0	0	✓	45
Присвоєння адресного номера об'єкту нерухомості (Н-14.2-14)	Управління містобудування, архітектури та кадастру	Містобудування, архітектура та охорона культурної спадщини	Зал 3	0	0	✓	50
Прийняття рішення про передачу у власність громадян 1- та 2-ох квартирних житлових будинків (Н-15-06)	Управління житлово-комунального господарства, благоустрою та екології	Житлово-комунальне господарство та екологія	Зал 3	0	0	✓	54
Прийняття квартир від підприємств, установ, організацій до комунальної власності	Відділ квартирної обліку та нерухомості	Квартирні питання	Зал 3	0	0	✓	37

Рисунок 4.34. Список послуг

На рис. 4.35 показана форма редагування однієї з послуг.

Форма вводу Напаштування публікації

Назва * Укладання договору оренди землі

Шифр послуги * A-25-12

Залп * Зал 1

Установа * Відділ земельних ресурсів

Категорія послуг * Земельні питання

Poslugy.gov.ua

Статус Опубліковано

Строк надання [Toggle Editor]

Місячний термін

Рисунок 4.35. Форма редагування послуги

Один із параметрів, за якими групуються послуги, є установи (або суб'єкти надання). На рисунку 4.36 наведений список, де можна вказати назву установи та її зображення.

Назва	Опублікований	ID
Архівний відділ	✓	1
Відділ Держгеокадастру у м. Тернополі Тернопільської області	✓	2
Відділ земельних ресурсів	✓	3
Відділ квартирного обліку та нерухомості	✓	4
Відділ обліку та фінансового забезпечення	✓	5
Головне управління Держгеокадастру в Тернопільській області	✓	6
Головне управління Держмагента у Тернопільській області	✗	7
Головне управління Держсанепідслужби у Тернопільській області	✓	8
Департамент екології та природних ресурсів Тернопільської обласної державної адміністрації	✓	9
Департамент культури, релігій та національностей Тернопільської обласної державної адміністрації	✓	10
Державна інспекція сільського господарства в Тернопільській області	✓	12
Інспекція державного архітектурно-будівельного контролю у Тернопільській області	✓	31
Служба у справах неповнолітніх та дітей	✓	13
Територіальне управління Держгірпромнагляду по Тернопільській області	✓	14

Рисунок 4.36. Список установ

Для зручності та пришвидшення пошуку послуги також розподіляються за категоріями (рис. 4.37).

Назва	Опублікований	ID
Архівні довідки	✓	6
Ветеринарні питання	✓	9
Житлово-комунальне господарство та екологія	✓	3
Земельні питання	✓	2
Інші послуги	✓	14
Квартирні питання	✓	4
Міграційні питання	✓	10
Містобудування, архітектура та охорона культурної спадщини	✓	1
Небезпечні вантажі та рух великогабаритного транспорту	✓	13
Підприємництво	✓	5
Пожезна безпека та охорона праці	✓	7
Реклама	✓	12
Санітарно-епідеміологічні питання	✓	8
Сім'я, молодь	✓	11

Рисунок 4.38. Список категорій послуг

Послуги центру надаються у трьох залах. За кожним закріплений власний перелік послуг, які у ньому можна отримати. На рис. 4.39 показана сторінка зі списком залів, де також можна переглянути статистичні дані щодо кількості позитивних та негативних відгуків про послуги, які цей зал надає.

Назва	Тривалість надання	Позитивні відгуки	Негативні відгуки	Опублікований	ID
Зал 1 - Сектор надання адміністративних послуг	30хв.	1	1	✓	1
Зал 2 - Сектор дозвілних процедур	30хв.	4	1	✓	2
Зал 3 - Сектор надання неадміністративних послуг	30хв.	1	0	✓	3

Рисунок 4.39. Список залів

На рисунку 4.40 зображена форма редагування одного зі залів. Тут можна вказати повну та коротку назву залу, графік роботи для запису в електронну чергу, тривалість надання однієї послуги у цьому залі. Відповідно до цієї кількості хвилин користувачі сайту будуть бачити різні проміжки часу, коли можна реєструватись на прийом.

Форма вводу Налаштування публікації

Назва * Зал 3

Повна назва * Зал 3 - Сектор надання неадміністративних послуг

Графік роботи

Дні	Час початку	Час закінчення
Понеділок	09 : 00	13 : 00
Вівторок	- : -	- : -
Середа	09 : 00	13 : 00
Четвер	14 : 00	16 : 00
П'ятниця	- : -	- : -
Субота	- : -	- : -
Неділя	- : -	- : -

Тривалість надання * 30 хв.

Статус Опубліковано

Рисунок 4.40. Форма редагування залу

Розроблений додаток відповідає вимогам, які ставилися замовником в процесі виконання науково-дослідного проекту («Веб-орієнтована програмна система підтримки діяльності Центру надання адміністративних послуг Тернопільської міської ради»).

Інтегрування інтелектуального модуля в існуючу систему дозволило підвищити ефективність функціонування ЦНАП шляхом скорочення часу опрацювання інформаційних запитів на етапі виконання відповідних послуг.

4.4. Експериментальні дослідження використання інтелектуального модуля

Для підтвердження ефективності запропонованих в роботі методів та програмних засобів було проведено ряд експериментальних досліджень. Їх суть в опрацюванні основних характеристик запитів користувачів до системи (час запиту, час формування відповіді, розмірність кешу) із підключенням інтелектуального модуля та без його використання.

Експериментальні дослідження проводилися на прикладі роботи Центру надання адміністративних послуг в місті Тернополі. Виходячи з структури інтелектуального модуля (метод оцінки подібності

адміністративних послуг та метод кластеризації документів) здійснювався вимір вище зазначених показників для кожної його складової.

На рисунку 4.41 представлено приклад реалізації алгоритму оцінки подібності адміністративних послуг.

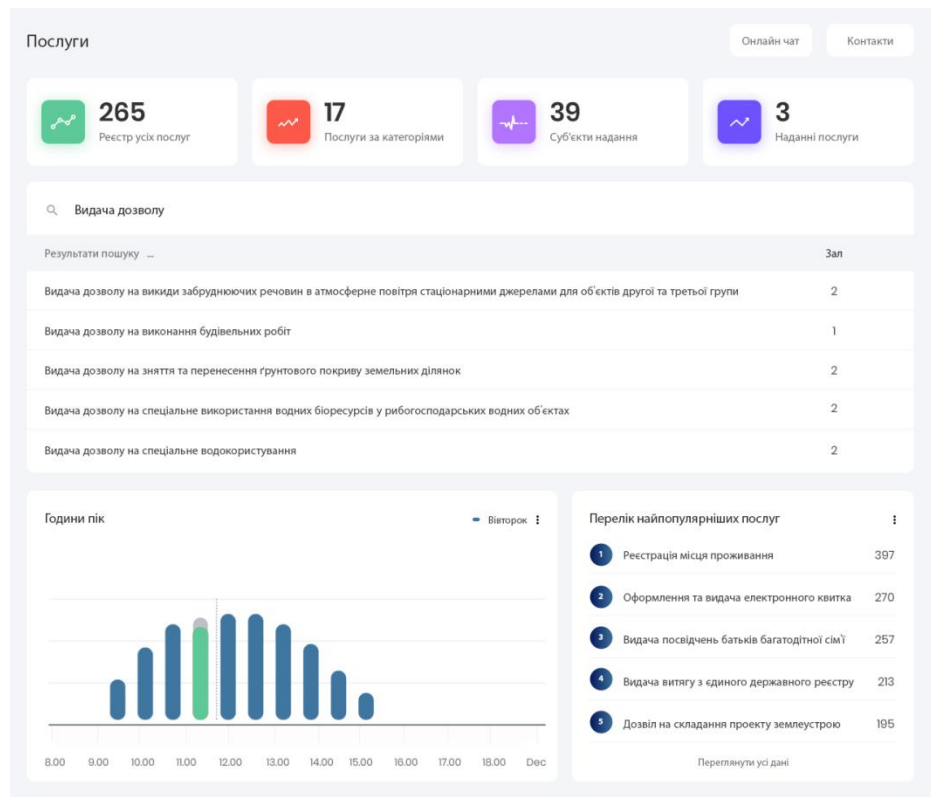


Рисунок 4.41. Приклад реалізації оцінки подібності адміністративних послуг

На рисунку 4.42 представлено приклад формування структури документів для адміністративної послуги «Видача дозволу на викиди забруднюючих речовин в атмосферне повітря стаціонарними джерелами для об'єктів другої та третьої групи».

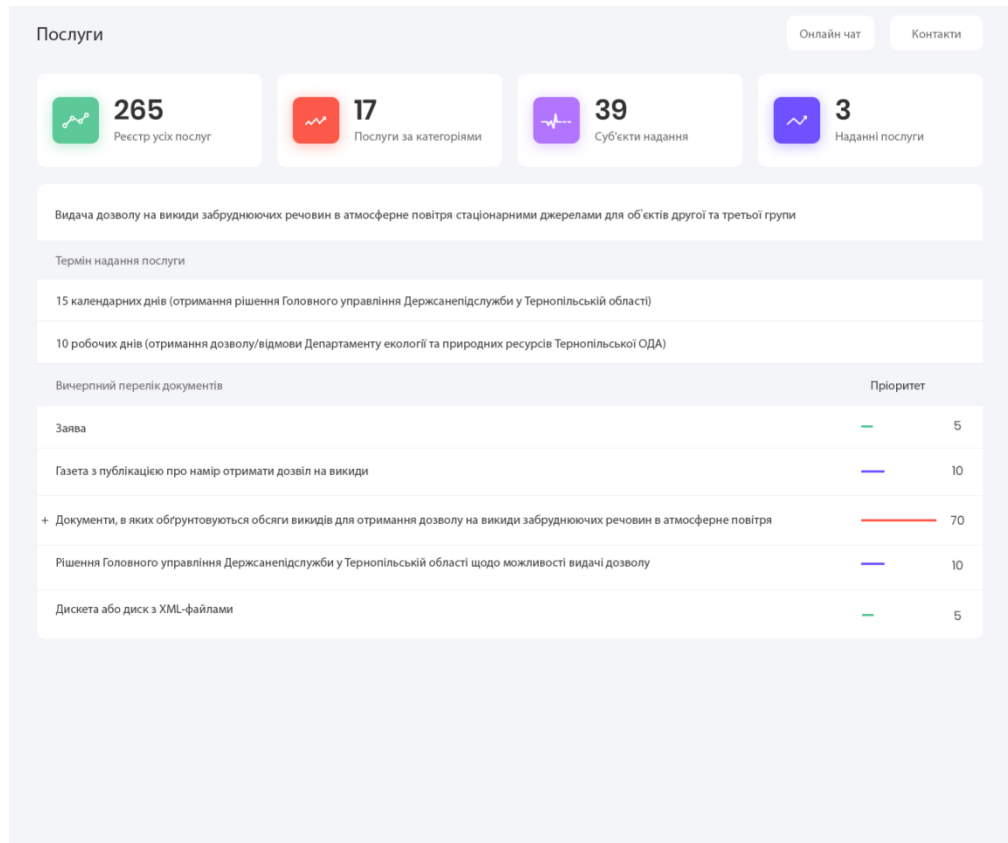


Рисунок 4.42. Інтерфейс інтелектуального модуля обробки інформації для системи центру надання адміністративних послуг

Отримана структура документів сформована з використанням інтелектуального модуля обробки інформації і є більш інформативною та простішою для пересічних користувачів.

Було проведено ряд експериментальних досліджень, які стосувалися оцінки ефективності застосування інтелектуального модуля. На рисунку 4.43 та 4.44 представлено порівняльну оцінку надання адміністративних послуг із підключенням інтелектуального модуля та з класичним використанням. Внаслідок цього, вдалося скоротити час надання для окремих категорій послуг від 15 до 50% за результатами експертного оцінювання.

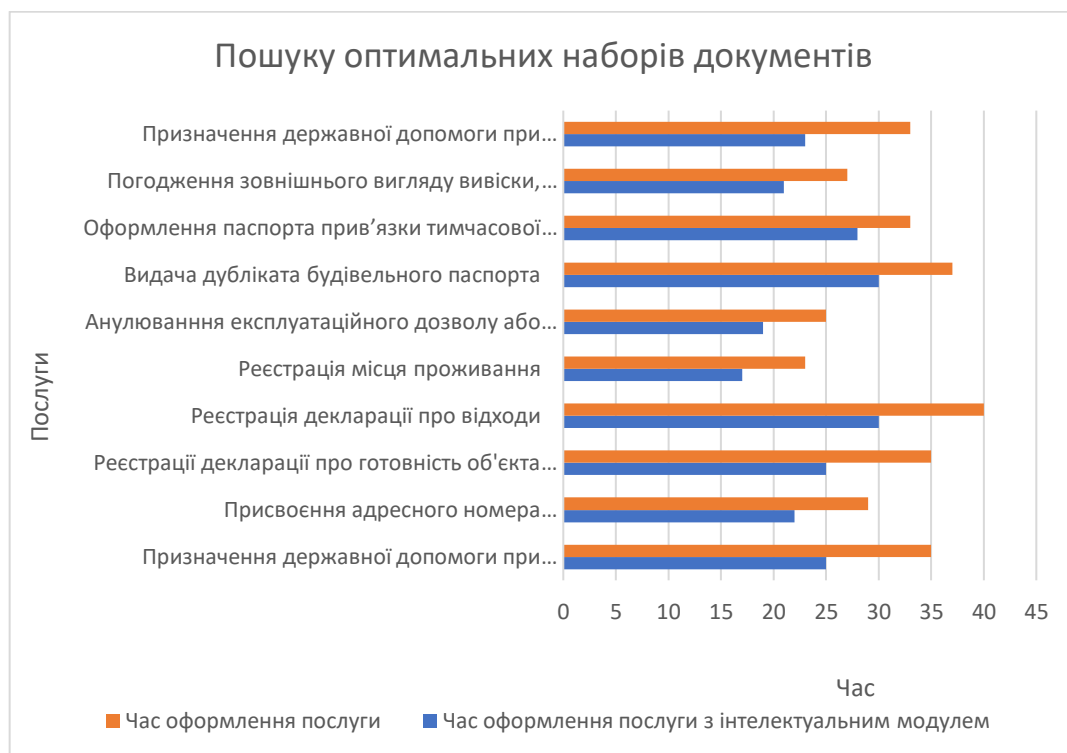


Рисунок 4.43. Порівняльна оцінка надання адміністративних послуг із підключенням інтелектуального модуля та класичним використанням (використовуючи метод пошуку оптимальних наборів документів)



Рисунок 4.44. Порівняльна оцінка надання адміністративних послуг із підключенням інтелектуального модуля та класичним використанням (використовуючи метод кластеризації документів).

Розроблена архітектура прикладної програмної системи центрів надання адміністративних послуг для рівня муніципалітетів, яка містить інтелектуальні модулі обробки інформації про надавані послуги та модулі оптимізації процесів виконання цих послуг за часом виконання. На тестових прикладах показано, що застосування створеного програмного забезпечення на їх основі зменшує час надання послуг з відгуку ресурсу від 15 до 50%.

Висновки до розділу 4

1. Представлено результати проектування програмного комплексу надання адміністративних послуг з інтеграцією інтелектуального модуля та його конструктивні особливості. Детально описано процес реалізації архітектури програмного комплексу, а також призначення інформаційних атрибутів при проектуванні системи зберігання інформації.

2. Представлено результати специфіки програмної реалізації системи із врахуванням структури запитів та обміну даними між інтелектуальним модулем і системою надання адміністративних послуг.

3. Модернізовано архітектуру веб-орієнтованої програмної системи, враховуючи специфіку інтеграції в її структурі інтелектуального модуля. Запропонована архітектура програмної системи реалізована із використанням сучасних інформаційних технологій, зокрема, використання PHP із фреймворком Zend та СУБД MySQL.

4. На основі експериментальних досліджень встановлено зменшення часу надання послуг від 15 до 50% внаслідок використання програмно-реалізованих методів: інтелектуального опрацювання інформації для оцінювання подібності онлайн послуг та автоматизованого пошуку оптимальних наборів документів в заданій предметній області з інтелектуальним модулем.

ВИСНОВКИ

В дисертаційній роботі розв'язано актуальну науково-прикладну задачу розробки математичного та програмного забезпечення підтримки веб-сервісів для надання адміністративних послуг шляхом впровадження засобів інтелектуалізації процесу виконання послуг та математичного моделювання ефективності функціонування веб-сервісів.

В процесі виконання дисертаційного дослідження встановлено наступні науково-практичні результати:

1. Проаналізовано існуючі методи та засоби організації веб-сервісів для надання адміністративних послуг. Встановлено, що відомі інформаційні системи надання адміністративних послуг не мають засобів інтелектуального аналізу даних та відповідних модулів їх реалізації.

2. Запропоновано та обґрунтовано показник ефективності функціонування веб-сервісів у вигляді динаміки відвідуваності користувачами та його математичну модель, яка, на відміну від існуючих, побудована на методах аналізу інтервальних даних та урахуванні циклічності процесів відвідуваності, що забезпечило гарантований щоденний прогноз кількості запитів веб-сервісу. Розроблена модель та метод її ідентифікації створили можливість планування та розподілу ресурсів організації щодо надання онлайн послуг. Для окремих сервісів це підвищило ефективність функціонування на 15%.

3. Розроблено метод інтелектуального опрацювання інформації з метою оцінювання подібності онлайн послуг, який ґрунтується на кластеризації документів, що оперують поняттями відстані між об'єктами-реквізитами. За результатами експертного оцінювання запропонований метод забезпечує зниження часу надання онлайн послуг на 20%.

4. Удосконалено метод автоматизованого пошуку оптимальних наборів документів в заданій предметній області з інтелектуальним модулем, що, за результатами експертного оцінювання, автоматизує і скорочує час процесу надання окремих категорій адміністративних послуг від 15 до 50%.

5. Розроблена архітектура прикладної програмної системи центрів надання адміністративних послуг для рівня муніципалітетів, яка містить інтелектуальні модулі обробки інформації про послуги, що надаються, та модулі оптимізації процесів за часом їх виконання. На тестових прикладах показано, що застосування створеного програмного забезпечення на їх основі зменшує час надання послуг від відгуку ресурсу з 15 до 50%.

6. Достовірність отриманих наукових результатів підтверджено апробацією запропонованих методів та архітектури прикладних програмних систем при вирішенні прикладних задач щодо підвищення ефективності функціонування веб-сервісів надання адміністративних послуг. Створене програмне забезпечення впроваджено в центрі надання адміністративних послуг у місті Тернополі, навчально-науковому центрі інформаційних технологій Західноукраїнського національного університету, Тернопільському освітньо-комунікаційному центрі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Interval Model of the Efficiency of the Functioning of Information Web Resources for Services on Ecological Expertise / M. Dyvak et. al. Mathematics. 2020. 8 (12). 2116. DOI: <https://doi.org/10.3390/math8122116>.
2. Дивак, М.П., Мельник, А.М., Папа, О.А. Математичне та програмне забезпечення інтелектуального модуля прикладних програмних систем для надання адміністративних послуг щодо проведення екологічної експертизи. Інформаційні технології та комп'ютерна інженерія. 2020. 49(3). С. 66–76. DOI: <https://doi.org/10.31649/1999-9941-2020-49-3-66-76>.
3. Підхід до математичного моделювання ефективності web-ресурсів / М. П. Дивак та ін. Оптико-електронні інформаційно-енергетичні технології. 2020. 38(2) С. 29–37. DOI: <https://doi.org/10.31649/1681-7893-2019-38-2-29-37>.
4. Features of Artificial Bee Colony Based Algorithm Realization for Parametric Identification Method of the Interval Discrete Dynamic Models / Y. Kedrin et. al. 2020. 10th International Conference on Advanced Computer Information Technologies (ACIT). P. 239–245. DOI: <https://doi.org/10.1109/ACIT49673.2020.9208916>
5. Software Module for Data Correctness and Completeness Control in the Academic Staff Performance Appraisal System / A. Pukas et. al. 2019 9th International Conference on Advanced Computer Information Technologies (ACIT). 2019. P. 277–280. DOI: <https://doi.org/10.1109/ACITT.2019.8779999>.
6. Інформаційна система підтримки надання послуг ЦНАП / А. В. Пукас та ін. Школа-семінар молодих вчених і студентів «Комп'ютерні інформаційні технології». 2019. С. 29–30.

7. Visitors Queue Management Optimization using Web System for Activity Support of the Administrative Services Center / O. Papa et. al. CEUR Workshop Proceedings. 2018. P. 187–190.
8. Clustering and interval analysis of heterogeneous data sample / M. Dyvak et. al. 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). 2017. P. 528–532. DOI: <https://doi.org/10.1109/STC-CSIT.2017.8098843> (дата звернення: 01.04.21).
9. Кедрін Є. С., Пукас А. В., Папа О. А. Особливості реалізації веб-орієнтованої системи підтримки діяльності центру надання адміністративних послуг. Сучасні комп'ютерні інформаційні технології: матеріали I Всеукр. конф. з міжнар. участю АСІТ'. 2017. С. 168–170.
10. Кедрін Є. С., Пукас А. В., Веремчук А. В., Папа О. А. Розширення функціоналу програмної системи підтримки діяльності центру надання адміністративних послуг Тернопільської міської ради. Сучасні комп'ютерні інформаційні технології: матеріали VI Всеукр. школи-семінару молод. вчених та студ. АСІТ'. 2016. С. 143–144.
11. Веремчук А. В., Пукас А. В., Порплиця Н. П., Папа О. А. Програмна система для моделювання об'єктів з розподіленими параметрами на основі інтервальних різницевих операторів. Сучасні комп'ютерні інформаційні технології: матеріали VI Всеукр. школи-семінару молод. вчених та студ. АСІТ'. 2016. С. 99–101.
12. Пукас А. В., Кедрін Є. С., Папа О. А. Веб-орієнтована програмна система підтримки діяльності центру надання адміністративних послуг Тернопільської міської ради. Сучасні комп'ютерні інформаційні технології: матеріали V Всеукр. школи-семінару молод. вчених та студ. АСІТ. 2015. С. 94–96.

13. «Регламент роботи Центру надання адміністративних послуг у місті Тернополі»: додат. до рішення виконавчого комітету від 17.04.2013 № 420.
14. «Положення про Центр надання адміністративних послуг міста Тернополя»: додат. до рішення Тернопільської міської ради від 15.02.2013 р. № 6/29/8.
15. The official website of the Government of Canada. URL: <http://canada.ca/> (дата звернення: 01.04.20).
16. Service-public. URL: <https://www.service-public.fr/> (дата звернення: 01.04.20).
17. Serwis Rzeczypospolitej Polskiej. URL: <https://gov.pl> (дата звернення: 01.04.20).
18. Дія - Державні послуги онлайн. URL: <https://diia.gov.ua/> (дата звернення: 01.04.20).
19. iGov – портал державних посуг. URL: <https://igov.org.ua/> (дата звернення: 01.04.20).
20. Центр надання адміністративних послуг міста Івано-Франківська. URL: <https://www.cnar.if.ua/> (дата звернення: 01.04.20).
21. Портал адміністративних послуг міста Луцька. URL: <https://ap.volyn.ua> (дата звернення: 01.04.20).
22. Рівненський регіональний центр надання адміністративних послуг. URL: <https://cnar.rv.ua> (дата звернення: 01.04.20).
23. Encyclopedia of virtual communities and technologies. Subhasish Dasgupta, editor. Idea Group Reference, USA, 2006.
24. Web-сервис. Энциклопедический словарь СМИ. <https://smi.academic.ru/7/WEB-сервис> (дата звернення: 01.11.2018).
25. С.В. Одиночкина. Учебное пособие Разработка WEB-сервисов в инфокоммуникационных системах 2013. 124 с.

26. Словарь компьютерных терминов <http://computer.slovaronline.com/> (дата звернения: 01.11.2018).
27. Сорокин, А.В. Проблемы интернета: Взаимодействие Web-сервисов-новый этап развития интернета. http://www.aselibrary.ru/datadocs/doc_754ge.pdf (дата звернения: 01.11.2018).
28. Г.Л. Багиев, И. В Успенский, В.И. Ченцов Интерактивные модели маркетинговых решений на виртуальных рынках. СПб: СПГУЭиФ, 2008. 152 с.
29. Hypertext Transfer Protocol - HTTP/1.1 <http://www.ietf.org/rfc/rfc2616.txt> (дата звернения: 01.11.2018).
30. Сиддикуи Б. Безопасность Web-сервисов <http://www.iso.ru/rus/document5972.phtml> (дата звернения: 08.11.2018).
31. Сычев А.В. Перспективные технологии и языки веб-разработки <http://www.intuit.ru/departement/internet/atlwebdev/> (дата звернения: 09.11.2018).
32. Фастовский Э.Г. Сервис-ориентированные технологии интеграции информации. Лекции. Харьков: НТУХПИ, 2011 – 50 с <http://www.twirpx.com/file/1112914/> (дата звернения: 01.11.2018).
33. Столлингс В. Компьютерные сети, протоколы и технологии Интернета. BHV, 2005 – 817 с.
34. Рябов В.А., Несвижский А.И. Современные web-технологии. М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006 – 432 с.
35. SOAP Version 1.2 Part 0: Primer (Second Edition) <http://www.w3.org/TR/2007/REC-soap12-part0-20070427/> (дата звернения: 01.12.2018).

36. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)
<http://www.w3.org/TR/2007/REC-soap12-part1-20070427/> (дата
звернення: 01.12.2018).
37. SOAP Version 1.2 Part 2: Adjuncts (Second Edition)
<http://www.w3.org/TR/2007/REC-soap12-part2-20070427/> (дата
звернення: 01.12.2018).
38. Сычев А.В. Web-технологии <http://www.intuit.ru/department/internet>
(дата звернення: 01.12.2018).
39. С. Лукошкин Блог Что такое CMS (движок) и почему это важно для
бизнес-сайта? [http://partizzan.ru/chto-takoe-cms-dvizhok-i-pochemu-eto-](http://partizzan.ru/chto-takoe-cms-dvizhok-i-pochemu-eto-vazhnodlya-biznes-sajta/)
[vazhnodlya-biznes-sajta/](http://partizzan.ru/chto-takoe-cms-dvizhok-i-pochemu-eto-vazhnodlya-biznes-sajta/) (дата звернення: 01.12.2018).
40. Кеннеди Билл, Муссиано Чак «HTML и XHTML. Подробное
руководство (HTML & XHTML. The Definitive Guide)». 2010. 310 с.
41. Шмитт К. «CSS. Рецепты программирования (CSS: Cookbook)». 2011.
256 с.
42. Гудман Д. JavaScript и HTML. Сборник рецептов. Для профессионалов.
СПб.: Питер, 2013. 207 с.
43. Daniel Woolston Pro Ajax and the .NET 2.0 Platform. М.: Инфра-М, 2017.
143 с.
44. Коггзолл Джон PHP 5. Полное руководство. М.: Диалектика, 2011.
45. Кузнецов М. В., Симдянов И. В., Голышев С. В. PHP 5. Практика
разработки Web-сайтов. СПб.: БХВ-Петербург, 2015.
46. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL
и JavaScript. СПб.: Питер, 2016. 496 с.
47. Колисниченко Д. Н. Самоучитель PHP 5. СПб.: Наука и Техника, 2015.
640 с.
48. Веллинг Л., Томсон Л. Разработка Web-приложений с помощью PHP и
MySQL. М.: Вильямс, 2018. 880 с.

49. Котеров Д. В., Костарёв А. Ф. PHP5. СПб: БХВПетербург, 2018. 1104 с.
50. Конверс Тим, Парк Джойс, Морган Кларк PHP и MySQL «Библия пользователя». «Диалектика». Москва. 2017. Глава 1: «Преимущества PHP».
51. PHP Fast & Easy Web Development. 2nd Edition. Premier Press. 2014
52. Олищук А. В., Чаплыгин А. Н. Разработка WEB приложений на PHP 5. Профессиональная работа Вильяме. 2006.
53. Маліцький О. С. Порівняльний аналіз сучасних систем для ведення статистики на веб-сторінках. ВНТУ. 2016. URL: <http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/10936/600.pdf?sequence=3> (дата звернення: 01.04.19).
54. Лукинский В.С., Цвиринько И. А., Малевич Ю. В. Логистика. СПб.: Изд-во СПбГИЭУ, 2000.
55. Гери Д., Хорстманн К. JavaServer Faces. М.: Вильямс, 2008. 576 с.
56. Мальтцер Кевин, Михальски Брент Разработка CGI-приложений на Perl. М.: Издательский дом «Вильяме», 2001. 400 с.
57. Стратегия продвижения сайтов. Исследования и статистика. URL: www.digits.ru/articles/promotion/strategy.html (дата звернення: 01.04.21).
58. Беседовський О. М., Захаров Д. С. Методи аналізу ефективності роботи сайту. Системи обробки інформації: зб. наук. праць. 2012. Вип. 8 (106): Проблеми і перспективи розвитку ІТ-індустрії. С. 2–4.
59. Руководство для веб-мастеров от G-oogle. URL: <https://developers.google.com/search/docs/advanced/guidelines/webmaster-guidelines?hl=ru> (дата звернення: 01.04.21).
60. SEO блог. URL: <https://netpeak.net/ru/blog/category/seo/> (дата звернення: 01.04.21).
61. Блог компании Aweb. URL: www.aweb.ua/seo-blog (дата звернення: 01.04.21).

62. Продвижение бизнеса в Интернет. URL: www.promodo.ua (дата звернення: 01.04.21).
63. Садовский А. Оценка эффекта от продвижения интернет-ресурсов. Стратегия продвижения сайтов в поисковых системах, 2002. С. 9 – 17.
64. Ашманов И., Иванов А. Оптимизация и продвижение сайтов в поисковых системах. СПб.: Питер, 2009. 400 с.
65. Севостьянов И. О. Поисковая оптимизация. Практическое руководство по продвижению сайта в Интернете. СПб.: Питер, 2010. 240 с.
66. Галустян А. Г. Разработка информационного и визуального дизайна сайта факультета ЭИ ХНЭУ. Збірник наукових праць студентів ХНЕУ. Х.: ХНЕУ, 2011. С. 72–76.
67. Менаске Д., Алмейда В. Производительность Web-служб. Анализ, оценка и планирование. СПб.: ООО «ДиаСофтЮП», 2013. 480 с.
68. Розенфильд Л., Морвиль П. Информационная архитектура в Интернете. М.: ООО «И.Д. Вильямс», 2016. 544 с.
69. Abed Abedniya, Sahar Sabbaghi Mahmoudi, The Impact of Social Networking Websites to Facilitate the Effectiveness of Viral Marketing, December 2010, International Journal of Advanced Computer Science and Applications, Vol. 1, No.6
70. Пелецишин А. М. Позиціонування сайтів у глобальному інформаційному середовищі. - Львів: Видавництво Національного університету “Львівська політехніка”, 2007.- 260с.
71. Иванов Д. В. Виртуализация общества. Социология и социальная антропология. – СПб: Изд. „Петербургское Востоковедение”. – 2000. – С. 96.
72. Халилов Д. Маркетинг в социальных сетях. Москва: Манн, Иванов и Фербер , 2013. – С. 240.

73. Ашманов И. С. Иванов А.А. Продвижение сайта в поисковых системах. Вильямс, 2007. – С. 304.
74. Лапыгин Ю. Н. Управление проектами: от планирования до оценки эффективности. М.: Омега-Л, 2008. – С. 252.
75. Митяй О.В. Проектний аналіз: Навч. посіб. К. : Знання, 2011. С. 311.
76. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. Wiley; 11 edition, 2013. – P. 1296.
77. Marshall P., Rhodes M., Todd B. Ultimate Guide to Google AdWords, 4th Edition. Entrepreneur Media, Inc., 2014. – P. 397.
78. Кузьмін О., Мельник О. Теоретичні та прикладні засади менеджменту. Інтелект-Захід, 2002. – С. 65-211.
79. Серов Ю. О., Пелешишин А. М., Кравець Р. Б. Методи аналізу ефективності веб-форумів. Вісник Нац. ун-ту “Львівська політехніка”: Інформаційні системи та мережі. – Львів: Вид-во Львів. політехніки, 2009. – № 653. – С.197-206.
80. Syerov Yu., Fedushko S., Trach O. Improving the virtual communication efficiency as a fundamental factor of successful marketing strategy. Collection of scientific articles “Prospects for development of education and science”, Academic Publishing House of the Agricultural University, Plovdiv, Bulgaria, 2016. – P. 290-293.
81. Dyvak M. Parameters Identification Method of Interval Discrete Dynamic Models of Air Pollution Based on Artificial Bee Colony Algorithm. In Proceedings of the 10th International Conference on Advanced Computer Information Technologies (ACIT). 2020. P. 130–135.
82. Pasichnyk N., Melnyk, A., Dobrovolska N. Management the website attendance based on the projected traffic. In Proceedings of the 12th International Conference on the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM). 2013. P. 277–277. (1)

83. Manhas, D. A Study of Factors Affecting Websites Page Loading Speed for Efficient Web Performance. *International Journal of Computer Sciences and Engineering*. 2013.
84. Ivanov A. Mathematical models for basic functioning processes of an information web portal. *Sistemy i Sredstva Inform.* 2010. P. 106–132. (2)
85. Bartuskova A. Soukal I. The Novel Approach to Organization and Navigation by Using All Organization Schemes Simultaneously. In *Proceedings of the Perspectives in Business Informatics Research*. Springer International Publishing Cham, 2016. P. 99–106. (4)
86. Ganji M. D., Rahmanzadeh A. Chapter 6 - Mathematical modeling and simulation. In *Smart Nanocontainers. Micro and Nano Technologies*. Elsevier, 2020. P. 89–102. (5)
87. Tong H. *Non-linear time series: a dynamical system approach*. Oxford University Press. New York, 1990. (6)
88. Dyvak M., Pukas, A. Criterion of Design of Experiments for Tasks of Decision Support Interval Model Creation. In *Proceedings of the 2005 IEEE Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. 2005. P. 495–497.
89. Bifurcation analysis for the Internet congestion / Duran G. et. al. In *Proceedings of the IEEE INFOCOM IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2019. P. 1073–1074.
90. Porplytsya N., Dyvak M., Spivak, I., Voytyuk I. Mathematical and algorithmic foundations for implementation of the method for structure identification of interval difference operator based on functioning of bee colony. In *Proceedings of the The Experience of Designing and Application of CAD Systems in Microelectronics*. 2015. P. 196–199.
91. Rastrigin L. A. *Adaptation of complex systems*. Riga: Zinatne, 1981. P. 375. ()

92. Karaboga, D. An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report - TR06. Technical Report. Erciyes University. 2005.
93. Karaboga D., Basturk B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*. 2007. P.459–471.
94. Karaboga D., Kaya E. Estimation of number of foreign visitors with ANFIS by using ABC algorithm. *Soft Computing*. 2019.
95. Karaboga D., Aslan S. Discovery of conserved regions in DNA sequences by Artificial Bee Colony (ABC) algorithm based methods. *Natural Computing: an international journal*. 2019. P. 333–350.
96. Karaboga D., Basturk B. On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*. 2008. P. 687–697.
97. Karaboga D. Artificial bee colony algorithm. *Scholarpedia*. 2010.
98. Dyvak M., Porplytsya N. Formation and Identification of a Model for Recurrent Laryngeal Nerve Localization During the Surgery on Neck Organs. *Advances in Intelligent Systems and Computing III: selected papers from the International conference on computer science and information technologies CSIT*. 2019. P. 391– 404.
99. Porplytsya N., Voytyuk I., Pukas A., Dyvak T. Identification the interval difference operators based on artificial bee colony algorithm in task of modeling the air pollution from vehicular traffic. *The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM'2017): proc. of the XIVth Intern. conf. Lviv*. 2017. P. 58– 62.
100. Порплиця Н. П. Порівняльний аналіз ефективності генетичного та «бджолиного» алгоритмів у задачі структурної ідентифікації інтервального різницевого оператора. *Інформаційні технології та комп'ютерна інженерія*. 2015. № 1. С. 55– 67.

101. Очеретнюк Н. П., М. П. Дивак Структурна ідентифікація інтервальної моделі процесу сушіння гіпсокартону. Вимірювальна та обчислювальна техніка в технологічних процесах . 2013. № 2. С. 211–217.
102. Бригілевич І. Діяльність ЦНАП та оцінка якості надання адміністративних послуг. Київ, 2017. 40 с.
103. Dorigo Marco, Stützle Thomas Ant Colony Optimization: Overview and Recent Advances. es. 10.1007/978-1-4419-1665-5_8.
104. Тарловский В. А., Хайрова Н. Ф. Использование семантико-ориентированного лингвистического процессора для добывания новых знаний из потока документов корпоративной информационной системы. Вестник Национального технического университета. Харьков: НТУ «ХПИ», 2010. № 67. С. 132–138.
105. Дубовой В. М., Краковецкий О. Ю., Глонь О. В. Метод оцінки подібності веб-сторінок. Оптико-електронні інформаційно-енергетичні технології. 2008. № 2 (16). С. 5–9.
106. Люгер Джордж, Ф. Искусственный интеллект: стратегии и методы решения сложных проблем / пер. с англ. 4-е издание. М.: Издательский дом «Вильямс», 2005. 864 с.
107. Мельник А. М., Миць О. З. Підвищення ефективності пошуку документів в інтернет із врахуванням подібності веб-сторінок. Сучасні комп'ютерні інформаційні технології: матеріали III Всеукр. школи-семінару молод. вчених і студ. АСІТ'2013, м. Тернопіль, 17-18 трав. 2013 р. / відп. за вип. М. П. Дивак. Тернопіль: ТНЕУ, 2013. С. 212.
108. Лисенко О. О., Кокітко Р. І. Модель аналізу неструктурованої інформації для побудови бази знань корпоративної інформаційної системи. Сучасні комп'ютерні інформаційні технології: матеріали III Всеукр. школи-семінару молод. вчених і студ. АСІТ'2013, м. Тернопіль,

- 17-18 трав. 2013 р. / відп. за вип. М. П. Дивак. Тернопіль: ТНЕУ, 2013. С. 211.
109. Formalization of Scientific Researches Results in Corporate Knowledge Bases As a Tool of Their Accumulation / М. Susla et. al. 10th International Conference on Advanced Computer Information Technologies (ACIT), Deggendorf, Germany, 2020. P. 488–491.
 110. Method for detection of non-relevant and wrong information based on content analysis of web resources / A. Kovbasistyi et. al. XIIIth International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH). Lviv, 2017. P. 154–156.
 111. Dorigo M. Optimization, Learning and Natural Algorithms. PhD Thesis, Dipartimento di Elettronica, Politecnico Di Milano, Italy. — 1992. — 140 p.
 112. Dorigo M., Maniezzo V., Colorni A. The Ant System: Optimization by a Colony of Cooperating Agents // IEEE Trans. On Systems, Man and Cybernetics. Part B. — 1996. — № 1. — Vol. 26. — P. 29—41.
 113. Bonavear E., Dorigo M. Swarm Intelligence: from Natural to Artificial Systems. — Oxford University Press, 1999. — 307 p.
 114. Dorigo M. Swarm Intelligence, Ant Algorithms and Ant Colony Optimization // Reader for CEU Summer University Course
 115. Ant Colony Optimization: [http://iridia. ulb. ac. be/~mdorigo/ACO/ACO. Html](http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.Html)
 116. Goss S., Aron S., Deneubourg J. L., Pasteels J. M. Self-Organized Shortcuts in the Argentine Ant // Naturwissenschaften. — 1989.— № 76.— P. 579—581.
 117. Bullnheimer B., Hartl R. F., Strauss C. A New Rank-Based Version of the Ant System: A Computational Study // Central

118. European Journal for Operations Research and Economics. — 1999. — № 1(7). — P. 25—38.
119. Dorigo M., Gambardella L. M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem //IEEE Trans. on Evolutionary Computation. — 1997. — № 1(1). — P. 53—66.
120. Stutzle T., Hoos H. H. MAX-MIN Ant System // Future Generation Computer Systems. — 2000. — № 8(16). — P. 889—914.
121. Reinelt G. The Traveling Salesman: Computational Solutions for TSP Applications. Lecture Notes in Computer Science. Vol. 840. — Berlin: Springer-Verlag. — 1994.
122. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность: Пер. с англ. — М.: Мир, 1985. — 512 с.
123. Reimann M. Ant Based Optimization in Good Transportation. PhD Thesis. University of Vienna. — Vienna, Austria. — 2002.— 149 p.
124. Socha K., Knowles J., Smples M. A MAX-MIN Ant System for the University Course Timetabling Problem. In Proc. of the Third International Workshop on Ant Algorithms «ANTS 2002». Lecture Notes in Computer Science 2463: Springer-Verlag, 2002.— P. 1—13.
125. Mariano C. E., Morales E. MOAQ: An Ant-Q Algorithm for Multiple Objective Optimization Problems. In Proc. of Genetic and Evolutionary Computation Conference (GECCO-99). USA, San-Francisco, 1999, Vol. 1. — P. 894—901.
126. De Campos L. M., Gamez J. A., Puerta J. M. Learning Bayesian Networks by Ant Colony Optimisation: Searching in Two Different Spaces // Mathware & Soft Computing. — 2002. — № 9.
127. Raspinelli J. M., Lopes H. S., Freitas A. A. Data Mining with an Ant Colony Optimization Algorithm // IEEE Trans. On Evolutionary Computation.

- Special issue on Ant Colony Algorithms. — 2002. — V. 6. — № 4. — P. 321—332.
128. Caro G. D., Dorigo M. Anet: a Mobile Agents Approach to Adaptive Routing // IRIDA – Universite Libre de Bruxelles. — Brussels, Belgium. Technical Report IRIDA 97-12. — 27 p.
 129. Наместников А. М., Ярушкина И. Г. Эффективность генетических алгоритмов для задач автоматизированного проектирования // Известия РАН. Теория и системы управления. — 2002. — № 2. — С. 127—133.
 130. Дейт К.Дж. Введение в системы баз данных / пер. с англ. 6-е изд. К.: Диалектика, 2015. 784 с.
 131. Адміністративні послуги: стан і перспективи реформування: зб. матеріалів / В. П.Тимошук та ін. К., 2015. 428 с.
 132. Тимошук В. Надання адміністративних послуг у муніципальному секторі: навч. посіб. для посадових осіб місцевого самоврядування / підгот. Асоціацією міст України. Київ: ТОВ «Підприємство «ВІ ЕН ЕЙ», 2015. 124 с.

ДОДАТОК А

Лістинг реалізації інтелектуального модуля

```

<?php

include_once('DB.class.php');
include_once('process.class.php');
include_once('languageProcessor.class.php');

class SearchEngine extends Process{

    public function __construct(){

    }

    public function search($keyword, $context, $network_ID, $peopleInfo){

        if($context == "People")
            return $this->searchPeople($keyword, $context, $network_ID,
$peopleInfo);
        else
            return $this->searchKnowledge($keyword, $context,
$network_ID);
    }

    public function searchPeople($keyword, $context, $network_ID,
$peopleInfo){

        $db= new DB("knowledgefile");

        $specificSearchStr= "";
        $networkStr= "";
        $openBr="";
        $closeBr="";

        if($peopleInfo){
            $nationality= $peopleInfo['1stNationality'];
            $country= $peopleInfo['1stCountry'];
            $city= $peopleInfo['1stCity'];
            $position= $peopleInfo['1stPosition'];
            $specialism= $peopleInfo['1stSpecialism'];
            $subSpecialism= $peopleInfo['1stSubSpecialism'];
            $company= $peopleInfo['1stCompany'];
            $college= $peopleInfo['1stCollege'];
            $specificSearchStr .= " and ";

            //print_r($peopleInfo);

            if($nationality != "selected")
                $specificSearchStr .= "D.nationality_ID='" .
$nationality . "' and ";
            if($country != "selected")
                $specificSearchStr .= "D.country_ID='" . $country . "'
and ";
            if($city != "selected")
                $specificSearchStr .= "D.city_ID='" . $city . "' and
";

```

```

        if($position != "selected")
            $specificSearchStr .= "D.position_ID='" . $position .
'' and ";
        if($specialism != "selected")
            $specificSearchStr .= "D.specialism_ID='" .
$specialism . "' and ";
        if($subSpecialism != "selected")
            $specificSearchStr .= "D.sub_specialism_ID='" .
$subSpecialism . "' and ";
        if($company != "selected")
            $specificSearchStr .= "D.company_ID='" . $company . "'
and ";
        else if($college != "selected")
            $specificSearchStr .= "D.college_ID='" . $college . "'
and ";

        $specificSearchStr = substr($specificSearchStr, 0,
strlen($specificSearchStr)-5);
    }

    if($network_ID){
        $openBr="(";
        $closeBr=")";
        $networkStr= " AND (
                        A.user_ID
                        IN (
                        SELECT F.user_ID
                        FROM user_network F
                        WHERE F.network_ID = '$network_ID'
                        )
                    )";
    }

    $query= "SELECT A.user_ID, B.*
            FROM user A, userinfo B, logininfo C
            WHERE " . $openBr . "(
            A.userinfo_ID = B.userinfo_ID
            AND B.userinfo_ID
            IN (
            SELECT D.userinfo_ID
            FROM userinfo D
            WHERE D.name LIKE (
            '%$keyword%'
            )" . $specificSearchStr . "
            )
            OR (
            A.userinfo_ID = B.userinfo_ID
            AND A.logininfo_ID = C.logininfo_ID
            AND C.logininfo_ID = (
            SELECT E.logininfo_ID
            FROM logininfo E
            WHERE E.email = '$keyword' )
            )" . $closeBr . $networkStr . "
            GROUP BY A.user_ID";

    //echo $query;
    $rows= $db->retrieveRows($query);
    return $rows;

```

```

}

public function indexedSearch($keyword, $network_ID) {
    $lang= new LanguageProcessor();

    $tokens= $lang->tokenize($keyword);

    $db= new DB("kf_corpus");

    $lexicon_IDs= array();

    for($i=0;$i<count($tokens);$i++){
        $word= $tokens[$i];
        $query= "select * from lexicon where word = '$word'";
        $lexicon= $db->retrieveRow($query);

        if($lexicon)
            $lexicon_IDs[]= $lexicon['lexicon_ID'];
    }

    $subQuery= "";
    for($i=0;$i<count($lexicon_IDs);$i++){
        $subQuery .= "lexicon_ID = '" . $lexicon_IDs[$i] . "' or ";
    }

    $subQuery = substr($subQuery, 0, strlen($subQuery)-3);

    $query= "select * from knowledge_index where network_ID =
'$network_ID' and ($subQuery)";
    $recs= $db->retrieveRows($query);

    // $query= "select distinct knowledge_ID from knowledge_index where
network_ID = '$network_ID' and ($subQuery)";
    // $recs= $db->retrieveRows($query);

    //Prioritize the knowledge by awarding points
    $distinctKnowledgeIDs= array();
    $distinctKnowledge= array();

    for($i=0;$i<count($recs);$i++){
        $knowledge_ID= $recs[$i]['knowledge_ID'];
        $hits= $recs[$i]['hits'];

        if(!$lang->isMember($knowledge_ID, $distinctKnowledgeIDs)){
            $distinctKnowledgeIDs[]= $knowledge_ID;
            $distinctKnowledge["$knowledge_ID"]['hits']= $hits;
            $distinctKnowledge["$knowledge_ID"]['wordCount']= 1;
        }
        else{
            $distinctKnowledge["$knowledge_ID"]['hits'] += $hits;
            $distinctKnowledge["$knowledge_ID"]['wordCount']++;
        }
    }

    for($i=0;$i<count($distinctKnowledgeIDs);$i++){
        $knowledge_ID= $distinctKnowledgeIDs[$i];
        $distinctKnowledge["$knowledge_ID"]['points']=
$distinctKnowledge["$knowledge_ID"]['wordCount'] * 10000 +
$distinctKnowledge["$knowledge_ID"]['hits'];
    }
}

```

```

//print_r($distinctKnowledge);

$sortedKnowledge= array();
$max= count($distinctKnowledgeIDs);
for($x=0;$x<$max;$x++){

    $maxKnowledge_ID= $distinctKnowledgeIDs[0];
    $maxPoints=
$distinctKnowledge["$maxKnowledge_ID"]['points'];
    $removeInd= 0;

    for($y=1;$y<count($distinctKnowledgeIDs);$y++){

        $knowledge_ID= $distinctKnowledgeIDs[$y];
        $points=
$distinctKnowledge["$knowledge_ID"]['points'];

        if($maxPoints < $points){
            $maxPoints= $points;
            $maxKnowledge_ID= $knowledge_ID;
            $removeInd= $y;
        }

    }

    $sortedKnowledge[$x]= $maxKnowledge_ID;
    array_splice($distinctKnowledgeIDs, $removeInd, 1);
}
//echo "<br><br>";
//print_r($sortedKnowledge);

$db= new DB("knowledgefile");

$prioritizedRecs= array();

for($i=0;$i<count($sortedKnowledge);$i++){
    $knowledge_ID= $sortedKnowledge[$i];
    $query= "SELECT A.user_ID, B.name as userName, C.*, D.*, F.*
            FROM user A, userinfo B, user_network_knowledge
C, knowledge D, network F
            WHERE A.userinfo_ID = B.userinfo_ID
            AND A.user_ID = C.user_ID
            AND C.knowledge_ID = D.knowledge_ID AND
D.knowledge_ID = '$knowledge_ID'
            AND C.network_ID = '$network_ID' .
            " AND F.network_ID = C.network_ID";
    //echo $query;
    $rec= $db->retrieveRow($query);
    $prioritizedRecs[]= $rec;
}
return $prioritizedRecs;
}

public function searchKnowledge($keyword, $context, $network_ID){

    $db= new DB("knowledgefile");
    $networkStr= "";
    $contextStr= " (";

    if($network_ID)

```

```

        $networkStr .= " AND C.network_ID = '$network_ID'";

        if($context == "Articles" || $context[0] == 1)
            $contextStr .= " E.knowledgeType = '1' OR ";
        if($context == "Q&A" || $context[1] == 1)
            $contextStr .= " E.knowledgeType = '2' OR ";
        if($context == "Discussions" || $context[2] == 1)
            $contextStr .= " E.knowledgeType = '3' OR ";

        $contextStr = substr($contextStr, 0, strlen($contextStr)-4);
        $contextStr .= ") AND ";

        $query= "SELECT A.user_ID, B.name as userName, C.*, D.*, F.*
                FROM user A, userinfo B, user_network_knowledge C,
knowledge D, network F
                WHERE A.userinfo_ID = B.userinfo_ID
                AND A.user_ID = C.user_ID
                AND C.knowledge_ID = D.knowledge_ID" . $networkStr .
                " AND D.knowledge_ID
                IN (

                SELECT E.knowledge_ID
                FROM knowledge E
                WHERE" . $contextStr . "(E.title LIKE (
                '%$keyword%'
                )
                OR E.content LIKE (
                '%$keyword%'
                ))
                )
                AND F.network_ID = C.network_ID";
        //echo $query;
        $rows= $db->retrieveRows($query);
        return $rows;

    }

    public function recordUserActions($args){

        if($args['actionCase'] == 1){
            if($args['actionType'] == "peopleSearched"){

                $searches= $this->retrieveUserGeneralAction($args['user_ID'], "peopleSearched");
                $searches++;
                $this->updateUserGeneralAction($args['user_ID'],
                "peopleSearched", $searches);
            }
            else if($args['actionType'] == "knowledgeSearched"){

                $searches= $this->retrieveUserGeneralAction($args['user_ID'], "knowledgeSearched");
                $searches++;
                $this->updateUserGeneralAction($args['user_ID'],
                "knowledgeSearched", $searches);
            }
        }
        else if($args['actionCase'] == 2){
            if($args['actionType'] == "peopleSearched"){

```

```

        $searches= $this->retrieveUserAction($args['user_ID'],
$args['network_ID'], "peopleSearched");
        $searches++;
        $this->updateUserAction($args['user_ID'],
$args['network_ID'], "peopleSearched", $searches);
    }
    else if($args['actionType'] == "knowledgeSearched"){

        $searches= $this->retrieveUserAction($args['user_ID'],
$args['network_ID'], "knowledgeSearched");
        $searches++;
        $this->updateUserAction($args['user_ID'],
$args['network_ID'], "knowledgeSearched", $searches);
    }
}

}

?>

<?php

include_once('DB.class.php');

abstract class DM {

    protected $source_db;
    protected $target_db;

    public function __construct($source, $target){
        $this->source_db= $source;
        $this->target_db= $target;
    }

    public function extract($columns, $table, $condition, $dataQty){

        $db= new DB($this->source_db);

        $query= "select $columns from $table where $condition";

        //echo $query;

        if($dataQty == 1){

            $element= $db->retrieveElement($query);
            return $element;
        }
        else if($dataQty == 2){
            $row= $db->retrieveRow($query);
            return $row;
        }
        else if($dataQty == 3){
            $rows= $db->retrieveRows($query);
            return $rows;
        }
    }
}

```

```

public function load($table, $columns, $records){

    $db= new DB($this->target_db);

    $strRows= "";

    for($i=0;$i<count($records);$i++)
        $strRows .= $records[$i] . ",";

    $strRows = substr($strRows, 0, strlen($strRows)-1);

    $query = "insert into $table " . $columns . " values " . $strRows;

    //echo $query;

    $db->insert($query);

}

public function generate_time_ID(){

    $db= new DB($this->target_db);

    $week= date("W");
    $month= date("m");
    $year= date("Y");

    $query = "insert into dmtime (week, month, year) values ('$week',
'$month', '$year')";

    //echo $query;

    $db->insert($query);

    $query= "select time_ID from dmtime order by time_ID desc limit
0,1";

    //echo $query;

    $time_ID= $db->retrieveElement($query);
    return $time_ID;
}

abstract protected function mine($type);

}

?>
class C4_5{

    private $rootSample;
    private $rootSampleCount;
    private $classes;
    private $classCount;
    private $nodes;
    private $edges;
    private $nodeCount;
    private $targetAttrName;

    public function __construct($sample, $classArr, $targetName){

```

```

        $this->rootSample= $sample;
        $this->rootSampleCount= count($sample);
        $this->classes= $classArr;
        $this->classCount= count($classArr);
        $this->nodes= array();
        $this->edges= array();
        $nodeCount=0;
        $this->targetAttrName= $targetName;
    }

    public function getNodes(){
        return $this->nodes;
    }

    public function getEdges(){
        return $this->edges;
    }

    private function getFreq($sample, $sampleCount){

        $freq= array(0,0,0);
        for($i=0, $j=0;$i<$sampleCount;$i++){

            $class= $sample[$i][$this->targetAttrName];

            $freq[$class]++;

        }

        return $freq;
    }

    private function getOneClass($freq){

        $emptyClassCount=0;
        for($i=0; $i<$this->classCount; $i++){
            if($freq[$this->classes[$i]]==0)
                $emptyClassCount++;
            else
                $class= $this->classes[$i];
        }

        if($emptyClassCount!=$this->classCount-1)
            $class= -1;

        return $class;
    }

    private function fewCases($sampleCount){

        $threshold= 1; //intval(0.1 * $this->rootSampleCount);
        if($sampleCount < $threshold)
            return true;
        return false;
    }

    private function getFreqClass($freq){

        $maxFreq= $freq[$this->classes[0]];
    }

```

```

    $freqClass= $this->classes[0];

    for($i=0; $i<$this->classCount; $i++){

        if($maxFreq<$freq[$this->classes[$i]]){
            $maxFreq= $freq[$this->classes[$i]];
            $freqClass= $this->classes[$i];
        }

    }
    return $freqClass;
}

private function getErrorVal($freq, $class){

    $errCount= 0;

    for($i=0; $i<$this->classCount; $i++){

        if($this->classes[$i] != $class)
            $errCount += $freq[$this->classes[$i]];

    }
    return $errCount;
}

private function entropy($freq, $s){

    $info= 0;

    for($i=0; $i<$this->classCount; $i++){

        $temp = $freq[$this->classes[$i]] / $s;

        if($temp != 0)
            $info += $temp * log($temp, 2);

    }

    $info *= -1;
    return $info;
}

private function subsetOf($sample, $sampleCount, $attrName, $attrVal){

    $subset= array();

    for($k=0; $k<$sampleCount; $k++){

        if($attrName == "age"){

            if($sample[$k]["$attrName"] > $attrVal-10 &&
$sample[$k]["$attrName"] <= $attrVal){
                $subset[]= $sample[$k];
            }

        }
        else{
            if($sample[$k]["$attrName"] == $attrVal){
                $subset[]= $sample[$k];
            }
        }
    }
}

```

```

    }
    }
    return $subset;
}

private function buildDecisionTree($sample, $parent_id, $predVal,
$attrs, $attrVals){

    $sampleCount= count($sample);

    $error=0;

    $freq= $this->getFreq($sample, $sampleCount);

    $oneClass= $this->getOneClass($freq);

    $edge= array();
    $edge['node1']= $parent_id;
    $edge['predVal']= $predVal;

    if($this->fewCases($sampleCount) || $oneClass!=-1 || count($attrs)
== 0){

        $node= array();
        $node['type']= 0;
        if($oneClass==-1 || count($attrs) == 0){
            $node['class']= $this->getFreqClass($freq);
            $node['error']= $this->getErrorVal($freq,
$node['class']);
        }
        else{
            $node['class']= $oneClass;
            $node['error']= 0;
        }
        $this->nodeCount++;
        $this->nodes["$this->nodeCount"]= $node;
        $edge['node2']= $this->nodeCount;
        $this->edges[]= $edge;

        return $node;
    }
    else{
        $node= array();
        $node['type']= 1;

        $bestGain= 0;
        $bestAttr= $attrs[0];
        $bestAttrInd=0;
        $sameGain= true;

        $info_T= $this->entropy($freq, $sampleCount);

        for($i=0;$i<count($attrs);$i++){

            $attrName= $attrs[$i];

            $attrValCount= count($attrVals["$attrName"]);

```

```

$info_Ti= 0;
for($j=0;$j<$attrValCount;$j++){
    $attrVal= $attrVals["$attrName"][$j];
    $subSample[$attrVal]= $this->subsetOf($sample,
$sampleCount, $attrName, $attrVal);
    $subSampleCount= count($subSample[$attrVal]);
    if($subSampleCount != 0){
        $freq_j= $this->getFreq($subSample[$attrVal], $subSampleCount);
        $temp= $this->entropy($freq_j,
$subSampleCount);
        $info_Ti += ($subSampleCount /
$sampleCount) * $temp;
    }
}

$gain= $info_T - $info_Ti;
if($bestGain < $gain){
    $bestGain= $gain;
    $bestAttr= $attrName;
    $bestAttrInd=$i;
}

if($bestGain != $gain)
    $sameGain= false;
}

if($sameGain){
    $node= array();
    $node['type']= 0;

    $node['class']= $this->getFreqClass($freq);
    $node['error']= $this->getErrorVal($freq,
$node['class']);

    $this->nodeCount++;
    $this->nodes["$this->nodeCount"]= $node;
    $edge['node2']= $this->nodeCount;
    $this->edges[]= $edge;

    return $node;
}
else{
    $node['predictor']= $bestAttr;
    $this->nodeCount++;
    $this->nodes["$this->nodeCount"]= $node;
    $nodeID= $this->nodeCount;
    $edge['node2']= $nodeID;
    $this->edges[]= $edge;

    $attrValCount= count($attrVals["$bestAttr"]);

```

```

        $subAttrs= $attrs;
        $subAttrVals= $attrVals;

        array_splice($subAttrs, $bestAttrInd, 1);
        array_splice($subAttrVals, $bestAttrInd, 1);

        for($i=0;$i<$attrValCount;$i++){

            $attrVal= $attrVals["$bestAttr"][$i];
            $subSample= $this->subsetOf($sample,
$sampleCount, $bestAttr, $attrVal);

            if(!$subSample){
                $childNode= array();
                $childNode['type']= 0;
                $childNode['class']= 0;           //$this-
>getFreqClass($freq);

                $childNode['error']= 0;
                $this->nodeCount++;
                $this->nodes["$this->nodeCount"]=
$childNode;

                $edge1= array();
                $edge1['node1']= $nodeID;
                $edge1['predVal']= $attrVal;
                $edge1['node2']= $this->nodeCount;
                $this->edges[]= $edge1;

            }
            else{
                $childNode= $this-
>buildDecisionTree($subSample, $nodeID, $attrVal, $subAttrs, $subAttrVals);
            }
        }
    }

    public function process_C4_5($attrs, $attrVals){
        $this->buildDecisionTree($this->rootSample, -1, "", $attrs,
$attrVals);
    }
}

?>
<?php

class FP_Growth{

    private $minSupport;
    private $fp_tree;
    private $fp_headerTable;
    private $cond_fp_tree;
    private $cond_headerTable;
    private $tree;
    private $nodeCount;
    private $headerTable;
    private $freqPatternSet;

```

```

public function __construct(){
    $this->minSupport=2;
    $this->nodeCount=0;
    $this->freqPatternSet=array();
}

public function getTree(){
    return $this->tree;
}

public function getHeaderTable(){
    return $this->headerTable;
}

public function getFreqPatternSet(){
    return $this->freqPatternSet;
}

private function getIndex($elem, $arr){
    for($i=0;$i<count($arr);$i++){
        if($elem == $arr[$i])
            return $i;
    }
    return -1;
}

private function getIndex2($elem, $objArr, $objAttrName){
    for($i=0;$i<count($objArr);$i++){
        if($elem == $objArr[$i]->$objAttrName)
            return $i;
    }
    return -1;
}

private function insertTree($itemSet, $parentNodeIndex, $is_cond_tree,
$cond_support){
    $first= $itemSet[0];
    $remSet= $itemSet;                                     // $remSet -> the
remaining portion of $itemSet or $sorted_NetworkIDSet
    array_splice($remSet, 0, 1);

    $exist= false;

    $parentNode= $this->tree[$parentNodeIndex];

    if($parentNode->hasChild()){
        for($i=0;$i<count($parentNode->childNodes);$i++){
            $childNodeIndex= $parentNode->childNodes[$i];

            $childNode= $this->tree[$childNodeIndex];
            if($childNode->itemID == $first){
                if($is_cond_tree)
                    $this->tree[$childNodeIndex]->support +=
$cond_support;
            }
        }
    }
}

```

```

        else
            $this->tree[$childNodeIndex]->support++;
            $nodeIndex= $childNodeIndex;
            $exist=true;
            break;
        }
    }
}

if(!$exist){
    $newNode= new Node(2);
    $newNode->nodeID= $this->nodeCount;
    $newNode->parentNodeID= $parentNodeIndex;
    $newNode->itemID= $first;

    if($is_cond_tree)
        $newNode->support = $cond_support;
    else
        $newNode->support= 1;

    $this->tree[$parentNodeIndex]->childNodes[]= $newNode-
>nodeID;

    $headerIndex= $this->getIndex2($newNode->itemID, $this-
>headerTable, "itemID");
    $this->headerTable[$headerIndex]->nodeLinks[]= $newNode-
>nodeID;

    $nodeIndex= $newNode->nodeID;

    $this->tree[$this->nodeCount++]= $newNode;
}

if(count($remSet)!=0)
    $this->insertTree($remSet, $nodeIndex, $is_cond_tree,
$cond_support);
}

public function construct_FP_Tree($dbScan1_results, $dbScan2_results){

    $itemSet= array();
    $support= array();

    //from the db first scan results, collect $itemSet, the set of
frequent items and their support counts

    $i= 0;
    for($j=0;$j<count($dbScan1_results);$j++) {

        $index= $this->getIndex($dbScan1_results[$j]['network_ID'],
$itemSet);
        if($index==-1){

            if(!$support[$i])
                $support[$i]=0;
            $support[$i]++;
            $itemSet[$i++]= $dbScan1_results[$j]['network_ID'];

        }
    }
}

```

```

        else{
            $support[$index]++;
        }
    }

    for($i=0;$i<count($itemSet);$i++){
        if($support[$i]<$this->minSupport){
            array_splice($itemSet, $i, 1);
            array_splice($support, $i, 1);
        }
    }

    $itemSet_support= array();
    for($i=0;$i<count($itemSet);$i++){
        $itemSet_support[$itemSet[$i]]= $support[$i];
    }

    //Sort $itemSet_support in support count descending order. arsort
    sorts the array in value (support) descending order
    arsort($itemSet_support);

    $i=0;
    while ($itemSet_support_elem = current($itemSet_support)) {
        $l1_itemSet[$i]= key($itemSet_support);
        $l1_support[$i]= $itemSet_support[key($itemSet_support)];
        $i++;
        next($itemSet_support);
    }

    //Initialize the root of an FP-tree as null
    $root= new Node(1); // 1 for root, 2 for internal
nodes, 3 for leaves
    $root->nodeID= 0;
    $root->parentNodeID= -1;
    //Initialize the tree
    $this->tree= array();
    $this->nodeCount=0;
    $this->tree[$this->nodeCount++]= $root;

    //Initialize the header table
    $this->headerTable= array();

    for($i=0;$i<count($l1_itemSet);$i++){
        $header= new Header($l1_itemSet[$i], $l1_support[$i]);
        $this->headerTable[$i]= $header;
    }

    //Use db second scan results
    for($x=0;$x<count($dbScan2_results);$x++) {
        $networkIDSet_str= $dbScan2_results[$x]['networks'];
        //echo "<br>". $networkIDSet_str . "<br>";
        $networkIDSet= array();
        $j=0;
        $ID=0;
        $step=1;

        for($i=0;$i<strlen($networkIDSet_str);$i++){
            if($networkIDSet_str[$i]!='-'){

```

```

                                $ID      =      $ID      *      $step      +
intval($networkIDSet_str[$i]);
                                $step *= 10;
                                }
                                else{
                                    $networkIDSet[$j++] = $ID;
                                    $ID=0;
                                    $step=1;
                                }
                                }
                                $networkIDSet[$j] = $ID;

                                //Select and sort the frequent items in each record or
transaction
                                for($i=0;$i<count($networkIDSet);$i++){
                                    $index=
                                $this->getIndex($networkIDSet[$i],
                                $l1_itemSet);

                                    if($index == -1)
                                        array_splice($networkIDSet, $i, 1);
                                }

                                if(count($networkIDSet)!=0){

                                    $sorted_networkIDSet= array();
                                    $k= 0;
                                    $isComplete= false;

                                    for($i=0;$i<count($l1_itemSet);$i++){

                                        if($isComplete)
                                            break;
                                        else{
                                            for($j=0;$j<count($networkIDSet);$j++){

                                                if($l1_itemSet[$i]==$networkIDSet[$j]){
                                                    $sorted_networkIDSet[$k++] =
                                $networkIDSet[$j];

                                                    if($k == count($networkIDSet))
                                                        $isComplete= true;
                                                    break;
                                                }
                                            }
                                        }
                                    }

                                    $this->insertTree($sorted_networkIDSet, 0, false, 0);
                                }
                                }
                                $this->fp_tree= $this->tree;
                                $this->fp_headerTable= $this->headerTable;
                            }

                            private function getParentItemIDs($tree, $nodeID){

                                $parentItemIDs= array();
                                $parentNodeID= $tree[$nodeID]->parentNodeID;

```

```

        while($parentNodeID != 0){           //while the node's parent is
not a root

            $itemID= $tree[$parentNodeID]->itemID;
            $parentItemIDs["$itemID"]= $itemID;

            $parentNodeID= $tree[$parentNodeID]->parentNodeID;
        }

        $parentItemIDs= array_reverse($parentItemIDs, true);

        return $parentItemIDs;
    }

    public function generate_Frequent_Patterns($ptnAlpha){
        $this->mine_Patterns($this->fp_tree,          $this->fp_headerTable,
$ptnAlpha);
    }

    public function mine_Patterns($cond_FP_Tree,      $cond_headerTable,
$ptnAlpha){

        for($i=0;$i<count($cond_headerTable);$i++){

            $ptnBeta= array();
            $ptnItemID= array();

            $itemID= $cond_headerTable[$i]->itemID;
            $support= $cond_headerTable[$i]->support;
            $ptnItemID["$itemID"]= $itemID;

            if(count($ptnAlpha)==0)
                $ptnBeta['pattern']= $ptnItemID;
            else
                $ptnBeta['pattern']=          $ptnItemID          +
$ptnAlpha['pattern'];
            $ptnBeta['support']= $support;

            //if(count($ptnBeta['pattern']) > 1)
            $this->freqPatternSet[]= $ptnBeta;

            $nodeLinks= $cond_headerTable[$i]->nodeLinks;

            //construct $ptnBeta's conditional pattern base
            $cond_ptn_set= array();

            for($j=0;$j<count($nodeLinks);$j++){

                $nodeID= $nodeLinks[$j];

                $cond_ptn= array();
                $parentItemIDs= $this->getParentItemIDs($cond_FP_Tree,
$nodeID);

                if(count($parentItemIDs)!=0){
                    $cond_ptn['pattern']= $parentItemIDs;
                    $cond_ptn['support']=      $cond_FP_Tree[$nodeID]-
>support;

```

```

        $cond_ptn_set[] = $cond_ptn;
    }
}

//Remove the item less than minimum support count

$itemSet = array();
$supportSet = array();

$y=0;
for($x=0;$x<count($cond_ptn_set);$x++){

    $itemIDs = $cond_ptn_set[$x]['pattern'];
    $support = $cond_ptn_set[$x]['support'];

    while ($itemIDs_elem = current($itemIDs)) {
        $itemID = key($itemIDs);

        $index = $this->getIndex($itemID, $itemSet);
        if($index == -1){

            if(!$supportSet[$y])
                $supportSet[$y] = 0;
            $supportSet[$y] += $support;
            $itemSet[$y++] = $itemID;

        }
        else{
            $supportSet[$index] += $support;
        }

        next($itemIDs);
    }
}

//Copy the items of $cond_ptn_set to $freq_cond_ptn_set if
the item has required minimum support
$freq_cond_ptn_set = array();
$freq_itemSet = array();
$freq_supportSet = array();

for($x=0;$x<count($itemSet);$x++){

    if($supportSet[$x] >= $this->minSupport){

        $includeItemID = $itemSet[$x];
        $freq_itemSet[] = $itemSet[$x];
        $freq_supportSet[] = $supportSet[$x];

        for($y=0;$y<count($cond_ptn_set);$y++){

            $itemIDs = $cond_ptn_set[$y]['pattern'];
            $support = $cond_ptn_set[$y]['support'];

            while ($itemIDs_elem = current($itemIDs)) {

                $itemID = key($itemIDs);
                if($itemID == $includeItemID){

                    $freq_cond_ptn_set[$y]['pattern'][] = $itemID;

```

```

    $freq_cond_ptn_set[$y]['support']= $support;
                                break;
                                }
                                next($itemIDs);
                            }
                        }
                    }

                if(count($freq_cond_ptn_set)!=0){

                    array_splice($freq_cond_ptn_set, 0, 0);
                //Compact the array to remove gaps or empty cells in between

                    //construct $ptnBeta's conditional FP-tree

                    //Initialize the root of an FP-tree as null
                    $root= new Node(1);                                // 1 for root, 2
for internal nodes, 3 for leaves
                    $root->nodeID= 0;
                    $root->parentNodeID= -1;
                    //Initialize the tree
                    $this->tree= array();
                    $this->nodeCount=0;
                    $this->tree[$this->nodeCount++]= $root;

                    //Initialize the header table
                    $this->headerTable= array();

                    for($x=0;$x<count($freq_itemSet);$x++){
                        $header= new Header($freq_itemSet[$x],
$freq_supportSet[$x]);
                        $this->headerTable[$x]= $header;
                    }

                    for($x=0;$x<count($freq_cond_ptn_set);$x++){

                        $this->
>insertTree($freq_cond_ptn_set[$x]['pattern'], 0, true,
$freq_cond_ptn_set[$x]['support']);
                    }

                    $this->cond_fp_tree= $this->tree;
                    $this->cond_headerTable= $this->headerTable;

                    if(count($this->cond_fp_tree)!=0)
                        $this->mine_Patterns($this->cond_fp_tree, $this->
>cond_headerTable, $ptnBeta);
                }
            }
        }

class Node{

    public $nodeID;

```

```

    public $parentNodeID;
    public $itemID;
    public $support;
    public $childNodes;
    public $type;

    public function __construct($type){
        $this->childNodes= array();
        $this->type= $type;
    }

    public function hasChild(){
        if(count($this->childNodes)==0)
            return false;
        return true;
    }

    public function isRoot(){
        if($this->type==1)
            return true;
        return false;
    }

    public function isLeaf(){
        if($this->type==3)
            return true;
        return false;
    }
}

class Header{

    public $itemID;
    public $support;
    public $nodeLinks;

    public function __construct($itemID, $support){
        $this->itemID= $itemID;
        $this->support= $support;
        $this->nodeLinks= array();
    }
}

```

?>

ДОДАТОК Б

DDL код бази даних системи зберігання інформації

```
CREATE TABLE IF NOT EXISTS `cnap_office_holidays` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `type` enum('recurrent','single','period') NOT NULL DEFAULT 'recurrent',
  `date_reccurent` date NOT NULL,
  `date_single` date NOT NULL,
  `date_period_1` date NOT NULL,
  `date_period_2` date NOT NULL,
  `published` tinyint(1) NOT NULL DEFAULT '0',
  `ordering` int(11) NOT NULL DEFAULT '0',
  `created` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `created_by` int(11) unsigned NOT NULL DEFAULT '0',
  `created_by_alias` char(100) NOT NULL DEFAULT '',
  `modified` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `modified_by` int(11) unsigned NOT NULL DEFAULT '0',
  `checked_out` int(11) NOT NULL DEFAULT '0',
  `checked_out_time` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  PRIMARY KEY (`id`),
  KEY `idx_checkout` (`checked_out`),
  KEY `idx_published` (`published`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE IF NOT EXISTS `cnap_office_services_categories` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `name` char(250) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `image` char(100) NOT NULL,
  `published` tinyint(1) NOT NULL DEFAULT '0',
  `ordering` int(11) NOT NULL DEFAULT '0',
  `created` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `created_by` int(11) unsigned NOT NULL DEFAULT '0',
  `created_by_alias` char(100) NOT NULL DEFAULT '',
  `modified` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `modified_by` int(11) unsigned NOT NULL DEFAULT '0',
  `checked_out` int(11) NOT NULL DEFAULT '0',
  `checked_out_time` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  PRIMARY KEY (`id`),
  KEY `idx_checkout` (`checked_out`),
  KEY `idx_published` (`published`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE IF NOT EXISTS `cnap_office_services` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `department_id` int(11) unsigned NOT NULL DEFAULT '0',
  `category_id` int(11) unsigned NOT NULL DEFAULT '0',
  `hall_id` int(11) unsigned NOT NULL DEFAULT '0',
  `code` char(50) NOT NULL,
  `name` varchar(500) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `description` text NOT NULL,
  `description_period` text NOT NULL,
  `description_result` text NOT NULL,
  `description_documents` text NOT NULL,
  `description_files` text NOT NULL,
  `duration_time` tinyint(3) unsigned NOT NULL,
  `derzhposluga` char(250) NOT NULL,
  `published` tinyint(1) NOT NULL DEFAULT '1',
```

```

`ordering` int(11) NOT NULL DEFAULT '0',
`created` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
`created_by` int(11) unsigned NOT NULL DEFAULT '0',
`created_by_alias` char(100) NOT NULL DEFAULT '',
`modified` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
`modified_by` int(11) unsigned NOT NULL DEFAULT '0',
`checked_out` int(11) NOT NULL DEFAULT '0',
`checked_out_time` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
PRIMARY KEY (`id`),
KEY `idx_checkout` (`checked_out`),
KEY `idx_published` (`published`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

ДОДАТОК В

Акти впровадженнь результатів дисертаційного дослідження

ЗАТВЕРДЖУЮ
 Начальник відділу «Центр надання
 адміністративних послуг»
 Тернопільської міської ради
 ПАНІЧЕВА І. Є.
 « 20 » 2020 р.
 * * * * *
 УКРАЇНА

АКТ

про впровадження результатів дисертаційної роботи
 Папи Олександра Андрійовича
 «Моделювання ефективності web-сервісів для надання адміністративних
 послуг»

У процесі розробки та впровадження прикладного програмного забезпечення центру надання адміністративних послуг Тернопільської міської ради використано результати дисертаційної роботи здобувача кафедри комп'ютерних наук Західноукраїнського національного університету Папи Олександра Андрійовича, а саме:

— показник ефективності функціонування web-сервісів у вигляді динаміки відвідуваності користувачами та його математичну модель, яка на відміну від існуючих, побудована на методах аналізу інтервальних даних та з урахуванням циклічності процесів відвідуваності, що забезпечило гарантований щоденний прогноз кількості запитів веб-сервісу;

— метод інтелектуального опрацювання інформації для оцінювання подібності он-лайн послуг, який на відміну від існуючих, ґрунтується на кластеризації документів;

— новий метод автоматизованого пошуку оптимальних наборів документів в заданій предметній області, що забезпечило створення інтелектуального програмного модуля, який автоматизує процес надання адміністративної послуги;

— архітектуру прикладної програмної системи центрів надання адміністративних послуг для рівня муніципалітетів, яка, на відміну від існуючих містить інтелектуальні модулі обробки інформації про надавані послуги, модулі оптимізації процесів виконання цих послуг за часом виконання.

Розроблені автором методи та прикладна програмна система безпосередньо впроваджені в ЦНАП, що у сукупності забезпечило підвищення ефективності функціонування веб-сервісів та скорочення часу надання адміністративних послуг у середньому на 15%.

Заступник начальника відділу
 «Центр надання адміністративних послуг»

А.В. ДЕМАКОВА

Завідувач сектору надання послуг
 виконавчих органів ради відділу
 «Центр надання адміністративних послуг»
 Адміністратор

О.О. ДУДЧЕНКО

ЗАТВЕРДЖУЮ

Директор навчально-наукового центру
інформаційних технологій
Західноукраїнського
національного університету
Ігор РОМАНЕНКО

« 16 » _____ 2020 р.

АКТ

про впровадження результатів дисертаційної роботи
Папи Олександра Андрійовича
**«Моделювання ефективності веб-сервісів для надання адміністративних
послуг»**

У процесі розробки та впровадження прикладного програмного забезпечення для доступу до інформаційних систем навчально-наукового центру інформаційних технологій із використанням веб-сервісів використано результати дисертаційної роботи здобувача ступеня доктора філософії Папи Олександра Андрійовича, а саме:

- метод інтелектуального опрацювання інформації для оцінювання подібності он-лайн послуг, який ґрунтується на кластеризації документів, що оперують поняттями відстані між об'єктами-реквізитами;
- показник ефективності функціонування web-сервісів у вигляді динаміки відвідуваності користувачами та його математичну модель, яка на відміну від існуючих, побудована на методах аналізу інтервальних даних та урахування циклічності процесів відвідуваності, що забезпечило гарантований щоденний прогноз кількості запитів веб-сервісу.

Розроблені автором методи безпосередньо впроваджені в Навчально-науковому центрі інформаційних технологій, що у сукупності забезпечило підвищення ефективності комунікації із зовнішніми додатками за рахунок скорочення часу надання послуг, а використання математичної моделі оцінки ефективності функціонування web-сервісів у вигляді динаміки відвідуваності користувачами дозволило спрогнозувати гарантований щоденний прогноз кількості запитів веб-сервісу.

Завідувач лабораторії розробки
та підтримки програмного забезпечення,
к.т.н., доцент



Андрій Мельник

ЗАТВЕРДЖУЮ

Директор тернопільської
міської громадської організації
«Тернопільський освітній
комунікаційний центр»
Сергій Возняк

13 / «30» 2020 р.



Номер: 22 від 30.10.2020

АКТ

про впровадження результатів дисертаційної роботи

Папи Олександра Андрійовича

«Моделювання ефективності веб-сервісів для надання адміністративних послуг»

У процесі розробки та впровадження прикладного програмного забезпечення для доступу до інформаційної системи тернопільської міської громадської організації «Тернопільський освітній комунікаційний центр» із використанням веб-сервісів використано результати дисертаційної роботи здобувача ступеня доктора філософії Папи Олександра Андрійовича, а саме:

- метод інтелектуального опрацювання інформації для оцінювання подібності он-лайн послуг, який ґрунтується на кластеризації документів, що оперують поняттями відстані між об'єктами-реквізитами;
- математична модель ефективності функціонування web-сервісів, яка прогнозує динаміку відвідуваності користувачів і на відміну від існуючих, побудована на методах аналізу інтервальних даних та з урахування циклічності та подібності процесів відвідуваності, що забезпечило гарантований щоденний прогноз кількості запитів веб-сервісу.

Розроблений автором метод, математична модель та прикладна програмна система безпосередньо впроваджені в Тернопільській міській громадській організації «Тернопільський освітній комунікаційний центр», що у сукупності забезпечило підвищення ефективності комунікації із зовнішніми додатками за рахунок скорочення часу доступу до сервісів, а використання математичної моделі оцінки ефективності функціонування web-сервісів дозволило спрогнозувати гарантований щоденний прогноз кількості запитів веб-сервісу.

ЗАТВЕРДЖУЮ

Перший проректор

Західноукраїнського

національного університету

к. ф. м. н. Микола ШИНКАРИК

«12» _____ 2020 р.

АКТ

про впровадження в освітній процес Західноукраїнського національного
університету результатів дисертаційної роботи
Папи Олександра Андрійовича
**«Моделювання ефективності веб-сервісів для надання адміністративних
послуг»**

Даний акт складений про те, що результати дисертаційної роботи здобувача ступеня доктора філософії Папи Олександра Андрійовича на тему: «Моделювання ефективності веб-сервісів для надання адміністративних послуг» використані в освітньому процесі факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету для студентів спеціальності «Інженерія програмного забезпечення»

При викладанні дисциплін «Архітектура та проектування програмного забезпечення», «Конструювання програмного забезпечення» та «Аналіз вимог до програмного забезпечення» розглядаються методи побудови архітектури прикладної програмної системи та засоби її розробки у випадку реалізації інтелектуальних програмних модулів, створених на основі поведінкової моделі мурашиної колонії. При викладанні дисципліни «Якість програмного забезпечення та тестування» у частині метрики та атрибути якості розглянуто методи дослідження ефективності веб-ресурсів та моделювання показників якості на основі інтервального аналізу даних.

Декан факультету комп'ютерних
інформаційних технологій,
д.т.н., професор



Микола ДИВАК

Завідувач кафедри
комп'ютерних наук
к.т.н., доцент



Андрій ПУКАС

Доцент кафедри
комп'ютерних наук,
к.т.н., доцент



Михайло ШПІНТАЛЬ