

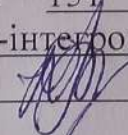
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

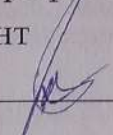
на тему:

«Автоматизована інформаційна система для розвідки
родовищ корисних копалин»

Виконав: студент 2 курсу, групи АКІТ-22мз,
спеціальність 151 – «Автоматизація та
комп'ютерно-інтегровані технології»

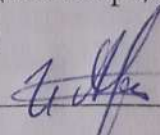
 Марія ЮРЧУК

Керівник: професор кафедри АІТ,
к.т.н., доцент

 Євген ПАЛАМАРЧУК

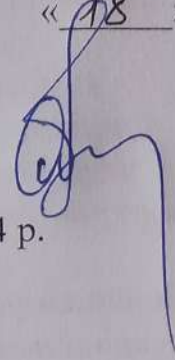
« 17 » червня 2024 р.

Опонент: доцент кафедри КН,
к.т.н., доцент

 Ігор АРСЕНЮК

« 18 » червня 2024 р.

Допущено до захисту
Завідувач кафедри АІТ
д.т.н., проф. Олег БІСІКАЛО


« 19 » 06 2024 р.

Вінницький національний технічний університет

Факультет інтелектуальних інформаційних технологій та автоматизації

Кафедра автоматизації та інтелектуальних інформаційних технологій

Рівень вищої освіти перший (бакалаврський)

Галузь знань 15 – Автоматизація та приладобудування

(шифр і назва)

Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології

(шифр і назва спеціальності)

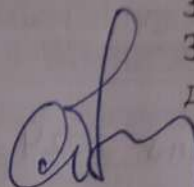
Освітня програма Інтелектуальні комп'ютерні системи

(назва освітньо-професійної програми)

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ

д.т.н., проф. Олег БІСІКАЛО.



12. березня 2024 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Юрчук Марії Сергіївні

(прізвище, ім'я, по батькові)

1. Тема роботи: «Автоматизована інформаційна система для розвідки родовищ корисних копалин»

Керівник роботи к.т.н., доц., професор кафедри АІТ Євген ПАЛАМАРЧУК
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджене протоколом №81 засідання кафедри АІТ від «11» березня 2024р.

2. Термін подання студентом роботи 13 червня 2024 р.

3. Вихідні дані до роботи: глобальна комп'ютерна мережа; операційна система Windows; підтримка усіх доступних браузерів; протокол передачі даних HTTPS; фреймворк клієнта jquery.js, оперативна пам'ять 2 GB.

4. Зміст текстової частини

- 1 Аналіз підходів до автоматизації розвідки родовищ корисних копалин в інформаційних системах;
- 2 Технології розробки модуля
- 3 Вибір технологій для реалізації
- 4 Розробка програмного модуля
- 5 Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)
- загальна структура інформаційного модуля
- діаграма варіантів використання;
- діаграма діяльності для серверної частини;

- діаграма послідовностей процесів;
- діаграма діяльності для користувацької частини;
- загальний вигляд інтерфейсу модуля;
- вигляд вікон редагування, друку та додавання.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	виконання прийняв
Спеціальна частина	к.т.н., професор кафедри АІТ Євген ПАЛАМАРЧУК	12.03.2024	17.05.2024
Економічна частина	к.е.н., професор кафедри ЕПтаВМ, Володимир КОЗЛОВСЬКИЙ	12.04.2024	19.05.2024

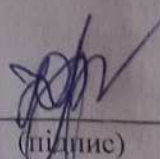
7. Дата видачі завдання 12 березня 2024 року

КАЛЕНДАРНИЙ ПЛАН

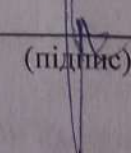
№	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1.	Вибір, узгодження та затвердження теми МКР	12.03.2024	26.03.2024	виконано
2.	Дослідження предметної області	27.03.2024	15.04.2024	виконано
3.	Обґрунтування методів реалізації та контролю	16.04.2024	22.04.2024	виконано
4.	Розробка структури та функціональної схеми	23.04.2024	18.05.2024	виконано
5.	Розробка програмного забезпечення	18.05.2024	08.05.2024	виконано
6.	Тестування та перевірка припущень	09.05.2024	12.05.2024	виконано
7.	Підтвердження економічної частини	12.05.2024	19.05.2024	виконано
8.	Оформлення матеріалів до захисту МКР	20.05.2024	05.06.2024	виконано

Студент

Керівник роботи


(підпис)

Марія ЮРЧУК
(прізвище та ініціали)


(підпис)

Євген ПАЛАМАРЧУК
(прізвище та ініціали)

АНОТАЦІЯ

УДК 004.896:622.8

Юрчук М. С. Автоматизована інформаційна система для розвідки родовищ корисних копалин. Магістерська кваліфікаційна робота зі спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології, освітня програма – Інтелектуальні комп'ютерні системи. Вінниця: ВНТУ, 2024. 151 с.

На укр.мові. Бібліогр.: 41 назв; рис.: 33; табл.: 13.

У магістерській роботі досліджено та розроблено автоматизовану інформаційну систему для розвідки родовищ корисних копалин. Основна мета роботи полягала у створенні ефективного інструменту для покращення процесів розвідки, зменшення витрат і підвищення точності отриманих даних. У ході дослідження було проведено аналіз існуючих систем, обґрунтовано вибір технологій для розробки користувацької та серверної частин, спроектовано архітектуру системи, реалізовано алгоритми обробки даних та виконано тестування розробленого програмного продукту. Для реалізації програмного модуля використовуються технології Oracle APEX, JQuery, KendoUI, PL/SQL та інші сучасні інструменти розробки. Результати роботи продемонстрували високу ефективність і комерційну привабливість системи, що підтверджується проведеними розрахунками економічного ефекту.

Ключові слова: автоматизовані інформаційні системи, розвідка родовищ, корисні копалини, геоінформаційні системи, обробка даних, аналіз родовищ.

ANNOTATION

UDC 004.896:622.8

Yurchuk M. S. Automated Information System for Mineral Exploration. Master's qualification work in the specialty 151 – Automation and Computer-Integrated Technologies, educational program – Intelligent Computer Systems. Vinnytsia: VNTU, 2024. 151 p.

In Ukrainian language. Bibliography: 41 titles; Figures: 33; Tables: 13.

This master's thesis investigates and develops an automated information system for mineral exploration. The main goal was to create an efficient tool to improve exploration processes, reduce costs, and increase data accuracy. The study involved analyzing existing systems, justifying the choice of technologies for developing the user and server parts, designing the system architecture, implementing data processing algorithms, and testing the developed software. Technologies such as Oracle APEX, JQuery, KendoUI, PL/SQL, and other modern development tools are used for the implementation. The results demonstrated high efficiency and commercial attractiveness of the system, confirmed by economic effect calculations.

Keywords: automated information systems, mineral exploration, minerals, geographic information systems, data processing, mineral analysis.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Введення в проблематику	7
1.2 Актуальність дослідження	9
1.3 Огляд та аналіз існуючих інформаційних систем розвідки родовищ	13
1.3.1 Історичний розвиток інформаційних систем розвідки родовищ	13
1.3.2 Основні компоненти інформаційних систем розвідки родовищ	15
1.3.3 Типи інформаційних систем у розвідці родовищ.....	16
1.3.4 Аналіз сучасних інформаційних систем розвідки родовищ та їх рішень.....	18
1.4 Висновки до розділу 1	28
2 ВИБІР ТЕХНОЛОГІЧНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ РОЗВІДКИ РОДОВИЩ КОРИСНИХ КОПАЛИН	30
2.1 Обґрунтування вибору технологій для розробки користувацької частини.....	30
2.2 Вибір інструментів для розробки серверної частини.....	36
2.3 Порівняння обраних інструментів для розробки з аналогами	40
2.4 Висновки до розділу 2	46
3 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ, ЇЇ ТЕХНОЛОГІЧНИХ МОДУЛІВ ТА КОМПОНЕНТІВ	47
3.1 Архітектура інформаційної системи.....	48
3.2 Аналіз функціональних вимог користувача.....	50
3.3 Проєктування бази даних	56
3.4 Розробка алгоритму функціонування інформаційної системи	66
3.5 Висновки до розділу 3	69
4 РЕАЛІЗАЦІЯ, ОПИС ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ РОЗВІДКИ РОДОВИЩ	71
4.1 Реалізація взаємодії клієнтської та серверної частини	71
4.2 Реалізація алгоритмів обробки даних у базі даних	75

	3
4.3 Реалізація користувацького інтерфейсу	78
4.4 Тестування розробки та аналіз результатів	94
4.5 Висновки до розділу 4	97
5 ЕКОНОМІЧНИЙ РОЗДІЛ	99
5.1 Технологічний аудит розробленої автоматизованої системи для розвідки родовищ корисних копалин	99
5.2 Розрахунок витрат на розроблення автоматизованої системи для розвідки родовищ корисних копалин	103
5.3 Розрахунок економічного ефекту від можливої комерціалізації розробки	107
ВИСНОВКИ.....	115
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	118
ДОДАТКИ.....	122
Додаток А (обов'язковий). Технічне завдання на магістерську кваліфікаційну роботу	123
Додаток Б (обов'язковий). Ілюстративна частина.....	126
Додаток В (обов'язковий). Лістинг програми.....	130
Додаток Г (обов'язковий). Протокол перевірки на плагіат.	152
Додаток Д (додатковий). Довідка про впровадження	153

ВСТУП

Актуальність. Розвідка родовищ є ключовим етапом у галузі гірничодобувної промисловості, який вимагає точного аналізу та оцінки природних ресурсів. В останні десятиліття значне зростання значення інформаційних систем у цьому процесі демонструє трансформацію традиційних методів розвідки в бік інтеграції цифрових технологій.

Сучасні програмні засоби, як-от GIS (географічні інформаційні системи), 3D моделювання, а також платформи з використанням штучного інтелекту та машинного навчання, відкривають нові можливості для геологічних досліджень та розвідки. Ці технології дозволяють здійснювати глибокий аналіз геологічної структури, прогнозувати потенціал родовищ й оптимізувати процеси видобутку природних ресурсів з максимальною ефективністю та мінімальними екологічними ризиками. [1]

Програми, такі як Petrel, Schlumberger та GPlates, надають геологам і розвідникам потужні інструменти для аналізу сейсмічних даних, моделювання геологічних процесів й управління даними родовищ. Застосування таких програм значно підвищує точність оцінки ресурсів, дозволяє виявляти нові потенційні родовища й сприяє більш раціональному плануванню видобувних робіт.

Інтеграція цифрових технологій з традиційними методами геологічних досліджень створює нову парадигму в розвідці корисних копалин, знижує витрати і збільшує шанси на успішне виявлення та експлуатацію родовищ. Окрім цього, сучасні програми дозволяють проводити комплексну екологічну оцінку та прогнозувати вплив видобувних процесів на довкілля, що вкрай важливо для сталого розвитку й відповідального користування природними ресурсами.

Впровадження й використання сучасних програмних рішень в інформаційних системах для розвідки родовищ корисних копалин стає не просто актуальним напрямком сталого розвитку, а й необхідною умовою для забезпечення технологічної й економічної ефективності, безпеки та екологічної

відповідальності в галузі.

Мета і задачі дослідження. Основною метою магістерської роботи є розробка автоматизованої інформаційної системи для ефективної розвідки родовищ корисних копалин. Для досягнення цієї мети в роботі поставлені наступні задачі:

1. Аналіз існуючих інформаційних систем у галузі геологічної розвідки та оцінка їх ефективності.
2. Визначення ключових вимог і функцій для розробки автоматизованої системи розвідки родовищ.
3. Розробка концепції інформаційної системи, що включає архітектуру, основні модулі та методи обробки даних.
4. Тестування і валідація функціоналу розробленої системи на прикладних задачах.

Об'єкт дослідження – процеси розвідки родовищ корисних копалин.
Предмет дослідження – автоматизована інформаційна система для розвідки родовищ корисних копалин.

Методи дослідження: в ході дослідження будуть використані методи системного аналізу, програмування, методи інженерії знань, моделювання процесів, зокрема побудова моделі знань на основі системи продукту, а також методи математичної статистики та обробки геологічних даних, системний аналіз, методи інженерії знань, зокрема побудова моделі знань на основі системи продукту, аналіз бізнес-процесів у виробництві, методи розробки клієнт-серверного програмного забезпечення.

Науково-технічним результатом магістерської роботи стане розробка програмного забезпечення для автоматизованої системи розвідки родовищ, яке буде спрямовано на підвищення точності та ефективності геологічних досліджень.

Практична цінність дослідження полягає у створенні інструментарію, котрий дозволить покращити процеси розвідки і видобутку корисних копалин, зменшити ризики й витрати при проведенні геологічних робіт. Розроблена в

межах даного дослідження автоматизована інформаційна система надасть експертам у галузі геології та розвідки нові можливості для аналізу та оцінки родовищ із значною економією часу та ресурсів.

Апробація результатів. Основні результати магістерської дипломної роботи прийшли апробацію на LIII Всеукраїнській науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації (м. Вінниця, 2024). За результатами магістерської дипломної роботи опубліковано тези доповідей у матеріалах LIII Науково-технічної конференції підрозділів Вінницького національного технічного університету [2].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Введення в проблематику

Протягом останніх десятиліть Україна переживає значні зміни у галузі розвідки родовищ корисних копалин, активізовані глобальними трендами та внутрішніми викликами. Цей період характеризується зростаючою потребою в автоматизації процесів з метою підвищення ефективності видобування та зниження екологічного впливу. В контексті глобалізації економіки та стрімкого технологічного розвитку, Україна зіштовхнулася з необхідністю адаптації до мінливих умов, включаючи зменшення запасів традиційних родовищ, зростання цін на енергоносії та зростання екологічних вимог. Відповідно, зростає потреба у розробці та інтеграції новітніх технологій, зокрема геоінформаційних систем для оптимізації розвідувальних робіт [3].

Окрім того, як відповідь на глобальні екологічні виклики та зобов'язання перед міжнародною спільнотою, Україна почала впроваджувати стандарти сталого розвитку у гірничодобувній індустрії, що вимагає відповідних змін у методології розвідки та видобування. Це зумовило потребу у глибокому аналізі існуючих підходів та розробці нових, більш ефективних і екологічно безпечних методів розвідки, котрі б дозволили з високою точністю оцінювати потенціал родовищ та мінімізувати ризики.

Значну увагу приділено дослідженню і впровадженню інноваційних технологічних рішень, зокрема, автоматизованих систем, які дозволяють здійснювати більш точний і ефективний аналіз геологічних даних. Впровадження цифрових технологій, таких як геоінформаційні системи (ГІС), тривимірне моделювання та дистанційне зондування, дозволило значно покращити якість та швидкість обробки геологічної інформації, сприяючи більш ефективному виявленню та оцінці родовищ [4].

З огляду на вищевказане, дослідження у сфері розвідки корисних копалин в Україні набуває нового виміру, зосереджуючись на інтеграції автоматизованих

технологій, оптимізації ресурсовикористання та мінімізації негативного впливу на довкілля. Ці зміни ставлять перед науковцями і фахівцями нові виклики, але одночасно відкривають перед галуззю нові можливості для розвитку й зростання.

Цифровізація процесів розвідки на родовищах є важливим напрямком розвитку гірничої промисловості, який спрямований на підвищення ефективності та зменшення негативного впливу на довкілля. Використання сучасних інформаційних технологій дозволяє значно покращити якість та точність проведення розвідувальних робіт, знижуючи при цьому витрати ресурсів та часу.

Одним із ключових аспектів інформатизації є використання геоінформаційних систем (ГІС). ГІС дозволяють збирати, аналізувати та візуалізувати геологічні дані, що значно полегшує процес прийняття рішень на етапі розвідки родовищ. Вони також дозволяють прогнозувати розташування корисних копалин з високою точністю, що сприяє ефективнішому використанню ресурсів та зменшенню впливу на навколишнє середовище [5].

Ключовим елементом інформатизації є використання автоматизованих систем для збору та обробки геологічних даних. Це дозволяє здійснювати більш точний аналіз геологічних структур, що в свою чергу сприяє виявленню нових родовищ та оптимізації процесів видобування.

Дистанційне зондування та тривимірне моделювання є важливими компонентами інформатизації процесів розвідки на родовищах. Вони дозволяють отримувати додаткові дані про геологічні структури та рельєф місцевості без необхідності прямого контакту з об'єктом дослідження [6].

У цілому, інформатизація процесів розвідки на родовищах сприяє підвищенню ефективності гірничих робіт, зменшенню негативного впливу на довкілля та забезпечує більш точні та достовірні результати розвідувальних робіт.

Автоматизація процесів розвідки на родовищах включає в себе різноманітні аспекти, що стосуються збору, аналізу, та використання даних для виявлення та оцінки корисних копалин. Розглянемо деякі з них більш детально:

– Сучасні технології збору даних: використання сучасних технологій, таких як лазерне сканування (LIDAR), супутникові знімки, аналізатори елементного складу, дозволяє отримувати великі обсяги геологічних даних з високою точністю та роздільною здатністю. Що у свою чергу допомагає здійснювати детальний аналіз ландшафту та структури родовищ.

– Моделювання геологічних процесів: використання комп'ютерних програм для тривимірного моделювання геологічних структур дозволяє прогнозувати розташування корисних копалин та оцінювати їхні запаси. Це допомагає зменшити ризики та витрати при проведенні розвідувальних робіт.

– Аналіз даних за допомогою штучного інтелекту: використання методів машинного навчання та штучного інтелекту дозволяє автоматизувати процес аналізу геологічних даних та виявлення закономірностей, що допомагає приймати більш обґрунтовані рішення [7].

– Інтеграція різних джерел даних: інформація з різних джерел, таких як геологічні карти, гравітаційні та магнітні дані, даних з буріння тощо, може бути інтегрована в єдину систему для отримання комплексного уявлення про структуру родовища.

– Візуалізація та аналіз даних: розвиток візуалізаційних технологій дозволяє представляти геологічні дані у вигляді тривимірних моделей та інтерактивних візуалізацій, що полегшує їх розуміння та аналіз.

– Безпека даних і кіберзахист: з огляду на важливість геологічних даних для промисловості, забезпечення їх безпеки й захист від кібератак стає все більш актуальною проблемою.

Зазначені аспекти цифровізації розвідки на родовищах постійно розвиваються, пристосовуючись до нових технологій та вимог галузі.

1.2 Актуальність дослідження

В умовах глобалізації економічного розвитку і стрімкого зростання населення планети, потреба в мінеральних ресурсах невпинно зростає. Це

викликає необхідність пошуку та розвідки нових родовищ корисних копалин, однак цей процес супроводжується рядом викликів, які необхідно вирішити для забезпечення ефективного та сталого використання природних ресурсів.

Зокрема, в Україні активізувалися зусилля з розвідки критичних мінералів, включаючи літій, кобальт та інші, з метою привернення інвестицій та розвитку. Вже розпочато аукціони з продажу ліцензій на розвідку, що залучили інтерес іноземних інвесторів. Програма може привабити інвестиції на суму близько \$10 мільярдів для розвитку близько 24 великих мінеральних родовищ [8].

Традиційні методи розвідки вимагають значних матеріальних і часових витрат. Дослідження родовищ є дорого вартісним процесом, який включає велику кількість польових робіт, збір та аналіз великої кількості даних. Крім того, ці методи часто є інтрузивними, тобто можуть негативно впливати на навколишнє середовище, порушуючи природний ландшафт та екосистеми.

Наступним важливим викликом є потреба в підвищенні точності розвідки. Існуючі методи часто не забезпечують достатньої деталізації та точності при визначенні місцезнаходження та обсягів родовищ, що призводить до нерентабельних інвестицій та неефективного використання ресурсів.

Екологічний аспект також стає дедалі важливішим у контексті глобальних змін клімату та зростаючої уваги до питань охорони довкілля. Вплив розвідувальних робіт на екосистеми, забруднення води та ґрунтів, втрата біорізноманіття – усі ці фактори потребують розробки та застосування більш екологічно безпечних та сталих методів розвідки.

Ще одним чинником, що впливає на галузь, є необхідність адаптації до швидких технологічних змін. Сучасний технологічний прогрес пропонує нові можливості для геологічної розвідки, включаючи застосування дистанційного зондування, геоінформаційних систем, штучного інтелекту та машинного навчання. Проте, інтеграція цих технологій вимагає значних інвестицій та розробки нових методик і підходів [4].

Таким чином, сучасні виклики в галузі розвідки родовищ вимагають комплексного підходу до їх вирішення, що включає вдосконалення

технологічної бази, оптимізацію процесів, зниження впливу на довкілля та врахування соціальних та економічних аспектів.

Однією з ключових технологій, яка змінює принципи розвідки родовищ, є дистанційне зондування. Ця технологія дозволяє вченим та інженерам отримувати важливу інформацію про земну поверхню без необхідності фізичної присутності на місцевості. Супутникові знімки та аерофотозйомка дозволяють виявляти потенційні родовища корисних копалин, оцінювати їх розміри та геологічні характеристики з високою точністю та ефективністю.

Геоінформаційні системи (ГІС) та інформаційні технології відіграють важливу роль у сучасній розвідці родовищ. Вони дозволяють збирати, зберігати, аналізувати та візуалізувати великі обсяги геологічних даних, полегшуючи процес прийняття рішень та планування розвідувальних робіт.

Штучний інтелект (ШІ) та машинне навчання відкривають нові можливості для аналізу геологічних даних. Застосування цих технологій може значно підвищити точність прогнозування розташування родовищ, аналіз їх потенційних запасів та оцінку ризиків. Алгоритми машинного навчання можуть обробляти та аналізувати великі обсяги даних швидше і ефективніше, ніж це робить людина.

Автоматизація розвідувальних робіт сприяє підвищенню безпеки й зниженню трудомісткості процесів. Автоматизоване буріння, зондування і вимірювання можуть проводитись з мінімальним втручанням людини, що зменшує ризик для здоров'я та життя працівників й підвищує точність отриманих даних.

Розвиток цифрових технологій та автоматизація не лише сприяють підвищенню ефективності й точності розвідки родовищ корисних копалин, але й допомагають зменшити їх негативний вплив на навколишнє середовище. Використання сучасних технологій є ключовим для подолання викликів, з якими стикається галузь із забезпечення сталого використання природних ресурсів.

Інтеграція автоматизованих систем підвищує ефективність розвідувальних робіт. Зменшення часу та витрат на пошук та аналіз потенційних родовищ

оптимізує управління ресурсами та зменшує виробничі витрати. Що сприятиме зниженню цін на кінцеву продукцію, стимулюючи економічне зростання та забезпечуючи конкурентні переваги на світовому ринку.

З іншого боку, покращення точності та ефективності розвідувальних робіт допомагає зменшити ризики, пов'язані з видобуванням корисних копалин, знижує кількість невдалих проєктів і, як наслідок, зменшення фінансових втрат. Економічна стабільність, досягнута завдяки ефективнішій розвідці, створює сприятливі умови для інвестування та розвитку інших секторів економіки.

На соціальному рівні, підвищення ефективності та безпеки розвідувальних робіт має позитивний вплив на забезпечення робочими місцями та зростання життєвого рівня в регіонах, де проводяться ці роботи. Крім того, зменшення негативного впливу на довкілля допомагає підтримувати екологічну рівновагу та забезпечує кращі умови життя для місцевих громад [9].

Важливим аспектом є також забезпечення енергетичної безпеки країни. Ефективна розвідка та раціональне використання ресурсів дозволяють державі зменшити залежність від імпорту енергоносіїв, сприяючи енергетичній незалежності та стабільності.

Так, автоматизація у галузі видобування корисних копалин в Австралії показала, що використання безпілотних вантажівок та дронів значно підвищує продуктивність й безпеку. Зокрема, випробування флоту з 10 автоматизованих вантажівок показало кращі показники безпеки та продуктивності порівняно з вантажівками, що керуються людьми. Також, зростає використання 4G і 5G технологій для забезпечення зв'язку та керування автоматизованим обладнанням. Розвиток автоматизації викликає зміну в структурі робочої сили, збільшуючи потребу у висококваліфікованих працівниках і високо функціональному програмному забезпеченні [10].

Загалом, автоматизація в галузі розвідки родовищ корисних копалин сприяє сталому розвитку економіки, покращенню соціальних умов та забезпеченню довготермінової екологічної та енергетичної безпеки.

1.3 Огляд та аналіз існуючих інформаційних систем розвідки родовищ

Технологічний прогрес суттєво трансформував геологорозвідувальну діяльність, відкриваючи нові горизонти для пошуку та аналізу родовищ корисних копалин. Завдяки цифровим технологіям, процеси розвідки стають більш ефективними, точними й менш затратними, що має значний вплив на весь ланцюжок вартості гірничодобувної промисловості.

Цифровізація та автоматизація процесів розвідки відкривають шлях до створення інтегрованих систем управління даними, що дозволяють ефективно зберігати, каталогізувати та аналізувати геологічну інформацію. Інформаційні системи допомагають об'єднати різноманітні дані, включаючи геологічні карти, результати буріння, лабораторні аналізи та інше, забезпечуючи швидкий доступ до необхідної інформації для прийняття обґрунтованих рішень.

1.3.1 Історичний розвиток інформаційних систем розвідки родовищ

Історичний розвиток інформаційних систем розвідки родовищ є свідченням значних змін у методах та підходах до пошуку корисних копалин. Початкові етапи розвідки родовищ ґрунтувались на прямих геологічних спостереженнях та простих фізичних методах. Із розвитком знань і технологій, методи розвідки ставали більш складними та ефективними.

На ранніх етапах, геологічна розвідка базувалася переважно на візуальній оцінці місцевості, фізичному відборі зразків та основних геофізичних методах, таких як гравіметрія та магнетометрія [11]. Ці методи дозволяли геологам отримати перші уявлення про будову надр, але мали обмежену точність та ефективність.

З розвитком технологій у середині ХХ століття почали використовуватися більш складні техніки, такі як сейсмічні дослідження. Цей метод дозволив виявляти геологічні структури на значних глибинах і з великою точністю. Однак, обробка та інтерпретація сейсмічних даних були довготривалими процесами, що

вимагали значних вкладень ресурсів.

Важливим етапом у розвитку інформаційних систем розвідки родовищ стало впровадження комп'ютерних технологій. Починаючи з 1970-х років, використання комп'ютерів дало можливість значно прискорити обробку даних, підвищити точність їх аналізу та полегшити візуалізацію геологічних структур. Комп'ютерні програми для моделювання родовищ та тривимірної візуалізації змінили підходи до планування та ведення розвідувальних робіт.

Сучасний етап розвитку інформаційних систем характеризується інтеграцією різноманітних технологічних рішень, включаючи дистанційне зондування, геоінформаційні системи (ГІС), штучний інтелект та машинне навчання.

До розгляду пропонуємо діаграму, що відображає еволюцію методів і систем розвідки родовищ з часом (рисунок 1.1).



Рисунок 1.1 – Еволюція інформаційних систем у георозвідці

Геоінформаційні системи забезпечують інтеграцію та систематизацію геологічних, геофізичних та екологічних даних, покращуючи управління даними та їх аналітику. Штучний інтелект та машинне навчання відкривають нові

можливості для автоматизації обробки даних та виявлення закономірностей, які складно ідентифікувати за допомогою традиційних методів.

Таким чином, інформаційні системи та технології відіграють вирішальну роль у сучасній геологорозвідувальній діяльності, забезпечуючи ефективність, точність та екологічну безпеку процесів пошуку та оцінки родовищ корисних копалин.

1.3.2 Основні компоненти інформаційних систем розвідки родовищ

Основні компоненти інформаційних систем розвідки родовищ формують їх структуру і функціональність, що дозволяє ефективно проводити пошук та оцінку мінеральних ресурсів. Розглянемо ці компоненти детальніше.

Апаратне забезпечення інформаційних систем розвідки родовищ включає широкий діапазон обладнання, яке необхідне для збору, зберігання та обробки даних. Це можуть бути геологічні інструменти, сейсмічні прилади, датчики дистанційного зондування, комп'ютерне обладнання та сервери для обробки великих обсягів даних. Висока продуктивність та надійність цього обладнання є ключовими для точної та ефективної роботи інформаційних систем.

Програмне забезпечення включає різноманітні програми та додатки для обробки геологічних даних, моделювання родовищ, візуалізації геологічних структур та управління базами даних. Спеціалізоване програмне забезпечення, таке як ГІС, дозволяє інтегрувати різноманітні дані та аналізувати їх в єдиному просторовому контексті.

Дані та їх обробка: дані є фундаментальним елементом будь-якої інформаційної системи розвідки. Вони можуть включати геологічні карти, сейсмічні записи, зразки порід, хімічні аналізи та іншу інформацію, зібрану під час розвідувальних робіт. Обробка цих даних включає їх класифікацію, аналіз та інтерпретацію, що дозволяє виявляти нові родовища, оцінювати їх потенціал та планувати подальшу діяльність.

Інтеграція інформаційних систем розвідки родовищ з іншими системами, такими як системи управління проєктами, управлінськими й логістичними

системами, сприяє покращенню координації, ефективності та продуктивності геологорозвідувальних робіт. Ця інтеграція дозволяє використовувати усю доступну інформацію для прийняття обґрунтованих рішень та оптимізації ресурсів.

У сукупності ці компоненти формують потужну систему, яка забезпечує комплексний підхід до розвідки родовищ, дозволяючи здійснювати пошук, оцінку та експлуатацію мінеральних ресурсів більш ефективно та екологічно безпечно.

1.3.3 Типи інформаційних систем у розвідці родовищ

Існує багато різних типів інформаційних систем, які використовуються в розвідці родовищ, кожна з яких виконує унікальні функції та є важливою частиною у геологорозвідування. Розглянемо найпоширеніші з них:

Геоінформаційні системи (ГІС): Геоінформаційні системи (ГІС) є одними з найважливіших інструментів у сучасній розвідці родовищ. Вони дозволяють збирати, зберігати, аналізувати та візуалізувати геологічні та географічні дані у формі карт, діаграм та звітів. ГІС забезпечують комплексний погляд на родовища, включаючи їх розміщення, геологічну будову, мінералогічний склад та інші важливі характеристики. Застосування ГІС сприяє оптимізації процесів пошуку та оцінки родовищ, дозволяючи ефективно планувати дослідницькі та видобувні роботи.

Системи дистанційного зондування: Системи дистанційного зондування використовуються для збору даних про земну поверхню та підземні структури без фізичного контакту з ними. Це може включати супутникові знімки, аерофотозйомку, радарні та лазерні сканування. Дані, отримані за допомогою дистанційного зондування, використовуються для картографування геологічних формацій, виявлення слідів мінералізації, аналізу рельєфу та інших геологічних завдань. Ці системи дозволяють проводити широкомасштабні геологічні дослідження з високою швидкістю та низькими витратами.

Системи управління даними родовищ (Database Management Systems, DBMS): Системи управління даними родовищ – це спеціалізовані бази даних, призначені для зберігання, управління та аналізу великих обсягів геологічної та експлуатаційної інформації. Вони забезпечують централізоване зберігання даних про родовища, включаючи результати буріння, геофізичні та геохімічні дослідження, звіти про розвідку та видобуток. DBMS сприяють ефективному доступу до даних, їхній інтеграції та використанню для прийняття рішень, планування та управління ресурсами родовищ.

Кожен з цих типів інформаційних систем має свої особливості та призначення, але разом вони створюють синергетичний інструментарій, який значно покращує якість та ефективність розвідувальних робіт, дозволяючи геологам та інженерам з вищою точністю виявляти та оцінювати родовища корисних копалин.

Переваги та недоліки різних систем унаочнені в таблиці 1.1.

Таблиця 1.1 – Переваги та недоліки інформаційних систем у розвідці родовищ

Тип інформаційної системи	Переваги інформаційної системи	Недоліки інформаційної системи
ГІС	Висока точність візуалізації, здатність інтегрувати різноманітні типи даних, підтримка прийняття обґрунтованих рішень	Висока вартість програмного забезпечення та обладнання, потреба у кваліфікованих фахівцях для ефективної роботи
Системи дистанційного зондування	Здатність швидко охоплювати великі території, можливість отримання даних у важкодоступних місцях	Обмежена роздільна здатність для певних типів даних, висока вартість супутникових знімків
DBMS	Забезпечення централізованого доступу до даних, підтримка складного аналізу та звітності	Комплексність налаштування та управління, висока вартість ліцензій та обслуговування

Приклади застосування на практиці:

- ГІС в розвідці нафти та газу: ГІС використовуються для інтеграції геологічних, геофізичних та бурінних даних для оцінки потенціалу нафтогазоносних басейнів, планування місць розташування нових свердловин та моніторингу експлуатаційної діяльності.

- Дистанційне зондування в мінеральній розвідці: Застосування дистанційного зондування для ідентифікації мінералізованих зон, картографування геологічних структур та визначення екологічних умов територій.

- DBMS у керуванні родовищами: Використання спеціалізованих баз даних для зберігання історичних даних буріння, ведення реєстру геологічних зразків, аналізу продуктивності родовищ і контролю за етапами видобутку [13].

1.3.4 Аналіз сучасних інформаційних систем розвідки родовищ та їх рішень

Сучасна галузь геологорозвідки характеризується широким застосуванням інформаційних систем, кожна з яких має свої унікальні функції, переваги та обмеження. Розглянемо приклади програмних продуктів за кожним з типів інформаційних систем, що дозволить краще зрозуміти, як вони застосовуються в галузі розвідки родовищ і які результати можуть бути отримані.

Розглянемо основні продукти представлені на ринку геоінформаційних систем: ArcGIS, QGIS та MapInfo Professional. Ці системи надають користувачам інструменти для збору, аналізу, візуалізації та управління геопросторовими даними.

ArcGIS, розроблена компанією Esri, є однією з найпотужніших і найбільш широко використовуваних геоінформаційних систем у світі. Ця платформа надає користувачам комплексний набір інструментів для збору, аналізу, управління та візуалізації геопросторових даних. ArcGIS широко застосовується у різних сферах, від містобудування та екології до розвідки природних ресурсів, таких як нафта і газ.

Завдяки можливостям платформи, користувачі можуть створювати детальні та інформативні карти, які допомагають у прийнятті обґрунтованих рішень. ArcGIS дозволяє проводити глибокий просторовий аналіз, що робить її незамінним інструментом для географічних досліджень та планування.

Платформа підтримує інтеграцію з великою кількістю джерел даних та систем, у тому числі з базами даних та іншими програмними рішеннями. Це дозволяє користувачам з легкістю комбінувати та аналізувати різноманітні дані для комплексного розуміння геопросторових задач. Однією з ключових особливостей ArcGIS є її здатність до 3D-візуалізації, що дозволяє користувачам створювати реалістичні тривимірні моделі місцевостей та аналізувати геопросторові дані у трьох вимірах.

Додатково, ArcGIS пропонує розширені можливості для мобільного збору даних та роботи у польових умовах, що значно полегшує процеси розвідки та дослідження на місцевості. Користувачі можуть використовувати ArcGIS Online для співпраці та обміну геоданими через хмару, що полегшує доступ до інформації та сприяє кращій координації між командами.

ArcGIS має широку глобальну спільноту користувачів, яка обмінюється досвідом, знаннями та інноваціями. Однак, необхідно враховувати, що висока функціональність і складність ArcGIS є складною для освоєння, а вартість програмного забезпечення та необхідного обладнання може бути завищеною для компаній.

QGIS є відкритим джерелом геоінформаційної системи, яка забезпечує користувачам потужний інструментарій для аналізу, обробки та візуалізації просторових даних без необхідності інвестування у дороге програмне забезпечення [12]. Ця система підтримує широкий спектр растрових та векторних форматів, що робить її сумісною з багатьма іншими геопросторовими додатками та сервісами.

QGIS вирізняється своєю спільнотою, яка постійно працює над вдосконаленням програми та розробкою нових плагінів, що розширюють її функціональність.

QGIS має інтуїтивно зрозумілий інтерфейс, який може бути легко налаштований під конкретні потреби користувача. Це робить її доступною, не тільки для досвідчених географів, але й для новачків у сфері ГІС. Програма дозволяє ефективно управляти геоданими, проводити складний просторовий аналіз та створювати високоякісні карти.

Однією з ключових переваг QGIS є її гнучкість: користувачі можуть адаптувати систему до своїх унікальних потреб завдяки великій кількості наявних плагінів та інструментів. Однак, слід враховувати, що хоча багато функцій доступні безпосередньо «з коробки», деякі складніші завдання можуть вимагати додаткового налаштування або програмування.

Незважаючи на свої переваги, QGIS може мати обмеження в порівнянні з комерційними системами, особливо коли йдеться про обробку великих обсягів даних або виконання дуже специфічних або новітніх аналітичних завдань. Крім того, через свій відкритий характер підтримка користувачів може бути не такою швидкою та організованою, як у платних програм. Однак, для багатьох завдань, зокрема для освітніх і дослідницьких цілей, QGIS є ідеальним вибором.

MapInfo Professional є потужним програмним забезпеченням геоінформаційної системи, яка розроблена компанією Pitney Bowes Software. MapInfo Professional широко використовується у різних сферах, від містобудування і транспорту до екології та розвідки родовищ. MapInfo надає користувачам інструменти для картографування, просторового аналізу, геокодування та створення звітів.

Однією з ключових переваг MapInfo Professional є його здатність інтегрувати дані з різних джерел та форматів, що робить його універсальним інструментом для аналізу та візуалізації геоданих. Користувачі можуть легко переглядати, редагувати та аналізувати просторові дані, ділитися ними з колегами та партнерами [14].

Програма забезпечує високу точність візуалізації даних завдяки своїм розширеним можливостям картографування. Користувачі можуть створювати карти різного ступеня складності, від простих схем до детальних тематичних

карт, які чітко відображають потрібну інформацію.

MapInfo Professional також пропонує розширені аналітичні функції, які дозволяють користувачам виконувати складний просторовий аналіз, включаючи буферизацію, накладання шарів, аналіз зони покриття та ін. Такі функції є корисними, наприклад, при плануванні міської інфраструктури або аналізі екологічних ризиків.

Програма має інтуїтивно зрозумілий інтерфейс, який робить її доступною для користувачів різного рівня кваліфікації. Хоча MapInfo Professional може вимагати певного часу для освоєння всіх функцій, багато користувачів відзначають її зручність та ефективність.

Однак, необхідно врахувати, що вартість ліцензії на MapInfo Professional є досить високою, що може бути обмежувальним фактором для малих компаній або індивідуальних користувачів. Крім того, хоча програма та її інструменти є дуже потужними, деякі специфічні аналітичні завдання можуть вимагати додаткових плагінів або інтеграції з іншими системами.

MapInfo Professional залишається одним із провідних рішень у сфері ГІС, яке відповідає потребам широкого кола фахівців, від геологів до містобудівників.

Важливо розуміти, як кожна з цих систем може відповідати конкретним потребам у розвідці родовищ (таблиця 1.2).

Таблиця 1.2 – Можливості геоінформаційних систем

Показники	ArcGIS	QGIS	MapInfo Professional
Візуалізація даних	Висока	Висока	Висока
Просторовий аналіз	Потужний	Потужний	Середній
Підтримка форматів даних	Широка	Широка	Широка
Інтеграція з іншими системами	Висока	Середня	Середня
Картографування	Детальне	Детальне	Детальне
Користувацький інтерфейс	Складний	Інтуїтивний	Інтуїтивний
Доступність	Платна	Безкоштовна	Платна
Спільнота та підтримка	Велика	Велика	Середня
Мобільні рішення	Так	Так	Так
Моделювання та аналіз	Високий рівень	Середній рівень	Середній рівень

Розглянемо детальніше найпопулярніші рішення серед систем дистанційного зондування.

ENVI є провідним програмним забезпеченням у сфері обробки та аналізу зображень, які отримані методом дистанційного зондування Землі. Розроблений компанією L3Harris Geospatial, ENVI використовується дослідниками, вченими та фахівцями в області геоінформатики по всьому світу для розшифрування супутникових знімків та аерофотознімків. Ця система має великий набір інструментів для виявлення, класифікації, аналізу та інтерпретації геопросторових даних. ENVI допомагає користувачам визначати характеристики земної поверхні, проводити моніторинг зміни довкілля, керувати природними ресурсами й виконувати інші завдання, де потрібна висока точність зображень [15].

Однією з особливостей ENVI є її здатність інтегруватися з іншими геоінформаційними системами та платформами даних, що дозволяє користувачам легко комбінувати та аналізувати дані з різних джерел. Крім того, ENVI надає інструменти для автоматизованого аналізу, які значно скорочують час обробки даних і підвищують продуктивність досліджень.

Програма постійно оновлюється, включаючи останні наукові досягнення й технологічні інновації, що робить її однією з найактуальніших у своєму класі. Незважаючи на високу функціональність, ENVI має дружній користувацький інтерфейс, що робить її доступною навіть для тих, хто не є експертом у галузі обробки зображень.

Однак, висока вартість і складність деяких аналітичних процесів можуть стати бар'єром для окремих користувачів або організацій з обмеженим бюджетом. Також, для ефективної роботи з ENVI потрібне розуміння основ дистанційного зондування та геопросторового аналізу. Незважаючи на це, ENVI залишається золотим стандартом у світі аналізу геопросторових даних, надаючи безпрецедентну точність та глибину аналізу для наукових досліджень і комерційних проєктів у всьому світі.

ERDAS IMAGINE є іншим важливим програмним рішенням у сфері

обробки зображень дистанційного зондування, розробленим компанією Hexagon Geospatial. Ця система забезпечує широкий спектр інструментів для роботи з геопросторовими даними, включаючи обробку супутникових знімків, аерофотозйомку, та аналіз рельєфу місцевості.

ERDAS IMAGINE використовується у різноманітних галузях: від моніторингу навколишнього середовища та сільського господарства до міського планування та оборони. Програма має можливість об'єднання різних типів даних, що дозволяє користувачам отримувати комплексне уявлення про досліджувані об'єкти та явища.

Однією з ключових особливостей ERDAS IMAGINE є її потужні засоби класифікації та аналізу даних, які дозволяють користувачам ефективно визначати та розрізняти різні природні та антропогенні об'єкти на знімках. Програма також забезпечує глибокий аналіз текстур та шаблонів, що робить її незамінною для детальних геологічних досліджень та розвідки ресурсів.

ERDAS IMAGINE підтримує автоматизацію рутинних завдань завдяки використанню скриптів та моделей, що зменшує час обробки даних та підвищує продуктивність користувачів. Крім того, програма має модульну структуру, що дозволяє додавати нові функції та інструменти відповідно до специфічних потреб проєкту.

До недоліків можна віднести: високий рівень складності й велику кількість функцій, які може бути складними для початківців та високу вартість ліцензії. Незважаючи на це, потужність та гнучкість ERDAS IMAGINE продовжують забезпечувати її популярність серед професіоналів у сфері дистанційного зондування та геопросторового аналізу.

Google Earth Engine є революційною платформою для аналізу даних дистанційного зондування та зображень Землі, яка надає науковцям, дослідникам та розробникам масштабний доступ до історичних та сучасних геопросторових даних. Ця потужна аналітична інструментальна платформа підтримує різноманітні додатки від моніторингу довкілля і зміни клімату до розвідки ресурсів та урбаністичного планування.

Однією з ключових переваг Google Earth Engine є її здатність швидко обробляти великі обсяги даних за допомогою хмарних технологій, що дозволяє користувачам виконувати складний геопросторовий аналіз без необхідності займатися інфраструктурою обчислень. Платформа забезпечує доступ до величезної колекції супутникових знімків, що оновлюються регулярно та охоплюють практично всю поверхню Землі [16].

За допомогою Google Earth Engine користувачі можуть легко виявляти зміни у ландшафтах, аналізувати тенденції зміни рослинності, водних ресурсів та інших природних атрибутів. Це робить платформу незамінним інструментом для екологічних досліджень та розробки стратегій збереження.

Незважаючи на свою потужність та гнучкість, Google Earth Engine має певні обмеження. Оскільки вона вимагає знань у програмуванні для розробки власних алгоритмів аналізу, користувачам без досвіду може бути важко використовувати всі її можливості. Крім того, доступ до деяких функцій платформи може бути обмежений для комерційного використання.

У таблиці 1.3 розглянемо можливості систем дистанційного зондування у порівнянні.

Таблиця 1.3 – Можливості систем дистанційного зондування

Показники	ENVI	ERDAS IMAGINE	Google Earth Engine
Обробка зображень	Висока	Висока	Висока
Аналіз змін	Потужний	Потужний	Дуже потужний
Класифікація даних	Висока	Висока	Висока
Інтеграція з ГІС	Так	Так	Так
Обробка великих наборів даних	Середня	Середня	Висока
Доступ до історичних даних	Обмежений	Обмежений	Дуже великий
Користувацький інтерфейс	Складний	Складний	Змінний
Підтримка форматів даних	Широка	Широка	Широка
Автоматизація процесів	Так	Так	Так
Спільнота та підтримка	Середня	Середня	Велика

Вивчимо детальніше існуючі рішення систем управління даними родовищ (DBMS).

Petrel E&P Software Platform, розроблена компанією Schlumberger, є однією з провідних платформ у сфері розвідки та виробництва нафти і газу. Petrel забезпечує інтегроване рішення, що охоплює всі етапи роботи від інтерпретації сейсмічних даних до моделювання резервуарів та планування буріння [17].

Ця платформа використовується фахівцями у всьому світі для аналізу геологічних даних та оптимізації розвідувальних та виробничих процесів. Petrel дозволяє користувачам легко інтегрувати дані з різних джерел, забезпечуючи точне й комплексне уявлення про підземні умови.

Особливістю Petrel є її високий рівень інтеграції та автоматизації, що дозволяє значно зменшити час від відкриття до виробництва. Система має інструменти для ефективного управління проєктами, від просторового моделювання до детального аналізу ризиків.

Petrel є флагманським продуктом у портфелі Schlumberger і постійно оновлюється, щоб відповідати сучасним вимогам і технологіям. Однак, потужність та складність системи вимагають від користувачів глибоких знань у галузі геології та інженерії.

Незважаючи на високу функціональність, Petrel може бути складним для освоєння новими користувачами, а висока вартість ліцензування робить її менш доступною для малих компаній або індивідуальних фахівців. Однак, для великих нафтогазових компаній, що шукають інтегроване рішення для управління своїми проєктами, Petrel є одним із популярних продуктів.

GVERSE Geomodeling від LMKR є геологічною моделюючою системою, яка призначена для полегшення та оптимізації процесів розвідки й розробки нафтових і газових родовищ. Цей програмний продукт дозволяє інтегрувати геологічні, геофізичні та інженерні дані для створення точних геологічних моделей та оптимізації рішень щодо буріння та виробництва.

GVERSE Geomodeling вирізняється своєю користувацькою адаптивністю та легкістю використання, що робить його доступним для геологів з низьким рівнем цифрових навичок. Програма має широкий набір аналітичних інструментів, що допомагають у рішенні різноманітних завдань, від оцінки

пористості та насиченості флюїдами до моделювання руху вуглеводнів.

Система GVERSE Geomodeling добре інтегрується з іншими програмами LMKR, зокрема з GeoGraphix, створюючи єдине робоче середовище для геологів й інженерів. Це дозволяє користувачам легко обмінюватися даними між модулями та ефективно використовувати всю доступну інформацію для прийняття обґрунтованих рішень.

Однією з переваг GVERSE Geomodeling є її гнучкість у виборі методик й алгоритмів моделювання, що дозволяє користувачам налаштовувати процеси аналізу з урахуванням специфіки конкретного родовища або проєкту. Програма також підтримує 3D-візуалізацію.

Водночас, GVERSE Geomodeling може мати певні обмеження для дуже складних або нетипових геологічних завдань, що вимагають унікальних або вузькоспеціалізованих рішень. Враховуючи широкий спектр можливостей, новим користувачам може знадобитися час на освоєння всіх функцій програми.

Незважаючи на це, GVERSE Geomodeling залишається цінним інструментом для компаній, що шукають ефективне та інтегроване рішення для геологорозвідувальної діяльності та розробки нафтогазових родовищ. Його здатність до інтеграції даних, спрощення аналізу та підтримка прийняття рішень роблять його незамінним помічником у плануванні та управлінні виробництвом нафти та газу.

Microsoft SQL Server є однією з провідних систем управління базами даних, яка використовується у різноманітних галузях для зберігання, обробки та аналізу великих обсягів даних. Завдяки вбудованим розширенням для геоданих, SQL Server є потужним інструментом для геопросторового аналізу та розвідки родовищ.

Ці розширення дозволяють користувачам зберігати, запитувати й управляти геопросторовими даними прямо всередині бази даних. Це може включати координати, карти, плани та інші типи просторової інформації, що є критично важливими для планування та аналізу в геології й розвідці.

SQL Server підтримує просторові дані у форматах, які відповідають

стандартам OGC (Open Geospatial Consortium), що забезпечує сумісність і можливість інтеграції з іншими геоінформаційними системами. Це робить його ідеальним вибором для компаній, які мають вимоги, щодо інтеграції й взаємодії між різними системами [18]. Крім того, SQL Server має розширені можливості для аналізу даних, включаючи машинне навчання і штучний інтелект.

Хоча Microsoft SQL Server є дуже потужною системою, вона може вимагати значних інвестицій як у ліцензування, так і в апаратне забезпечення. Також необхідно мати досвід роботи з базами даних і розуміння SQL для ефективного використання цієї системи.

Попри це, для великих компаній, які вже використовують інфраструктуру Microsoft або мають великі обсяги геопросторових даних, SQL Server з розширеннями для геоданих пропонує рішення, яке може бути інтегроване з іншими корпоративними системами та даними. Він забезпечує високий рівень безпеки, надійності та продуктивності, що є критично важливими для управління великими обсягами чутливої або критичної інформації.

Порівняємо можливості розглянутих систем управління даними родовищ (таблиця 1.4).

Таблиця 1.4 – Можливості систем управління даними родовищ (DBMS)

Показники	Petrel E&P Software Platform	GVERSE Geomodeling	Microsoft SQL Server (з розширеннями для геоданих)
Інтеграція даних	Висока	Висока	Висока
Геологічне моделювання	Висока	Висока	Обмежена
Аналіз родовищ	Висока	Висока	Середній
Візуалізація даних	Потужна	Потужна	Середня
Підтримка форматів даних	Широка	Широка	Широка
Управління проєктами	Високий рівень	Високий рівень	Середній рівень
Просторовий аналіз	Висока	Висока	Висока
Користувачський інтерфейс	Складний	Інтуїтивний	Складний
Доступність	Платна	Платна	Платна
Спільнота та підтримка	Велика	Середня	Велика

Кожен з цих програмних продуктів має свої унікальні функції та спеціалізації, але всі вони призначені для підвищення ефективності та точності розвідувальних робіт. Вибір певного програмного продукту залежить від конкретних потреб проєкту, наявних даних та бюджету.

1.4 Висновки до розділу 1

В процесі аналізу наявної літератури та статистичних даних було встановлено, що використання автоматизованих інформаційних систем у сфері розвідки та видобування корисних копалин є ефективним засобом підвищення продуктивності та економічної ефективності галузі. Автоматизація процесів дозволяє значно скоротити час на розвідку, знизити витрати на видобуток, а також підвищити точність і надійність отриманих даних. Це сприяє зменшенню кількості помилок, збільшенню обсягів ресурсів, що видобуваються, а також оптимізації використання природних ресурсів.

Активне впровадження інноваційних технологій та автоматизованих систем у галузі може сприяти залученню інвестицій та розвитку нових родовищ. З огляду на світові тенденції, Україна має значний потенціал для розвитку в даній сфері, що вимагає подальших досліджень, інвестицій та впровадження сучасних технологій для сталого розвитку галузі.

У рамках дослідження було розглянуто популярні програмні продукти, які застосовуються для автоматизації розвідки і видобутку корисних копалин. Серед них, програми для 3D моделювання геологічних структур, системи збору та аналізу геофізичних даних, а також програмні рішення для планування видобувних робіт та управління ресурсами родовища. Використання цих програмних продуктів демонструє зростаючу тенденцію до цифровізації галузі та підкреслює значення інформаційних технологій у підвищенні ефективності розвіdkово-видобувних процесів. Впровадження цих технологій не лише сприяє підвищенню точності геологічних прогнозів та оптимізації робочих процесів, але й відкриває нові можливості для розвитку галузі, зокрема, через збільшення

виявлених запасів корисних копалин та підвищення безпеки видобувних операцій. Таким чином, дослідження і впровадження новітніх програмних продуктів стають невід'ємною частиною стратегії розвитку галузі, яка спрямована на зростання продуктивності та економічної ефективності.

У зв'язку з цим, ключовими пріоритетами для України та світу у контексті розвідки та видобування корисних копалин є інтеграція автоматизованих систем, подальше дослідження та адаптація новітніх технологій, а також формування сприятливого інвестиційного клімату. Враховуючи це, дослідження та впровадження автоматизованих інформаційних систем у сфері розвідки та видобутку корисних копалин повинні стати важливими складовими розвитку галузі.

2 ВИБІР ТЕХНОЛОГІЧНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ РОЗВІДКИ РОДОВИЩ КОРИСНИХ КОПАЛИН

Вибір технологічних засобів є одним із ключових елементів успішного розвитку проєкту в сфері розвідки родовищ корисних копалин. Він впливає на продуктивність, економічну ефективність, надійність, безпеку, масштабованість і здатність до інновацій. Ретельний аналіз та обґрунтований вибір технологій забезпечить оптимальні результати й сприятиме успішній реалізації проєкту.

Вибір технологічних засобів для реалізації інформаційної системи для автоматизації розвідки родовищ корисних копалин, зокрема, включає розробку на рівні frontend (клієнтська частина) та backend (серверна частина). Основна відмінність між ними полягає в тому, що frontend взаємодіє безпосередньо з користувачами, надаючи інтерфейс для візуалізації та взаємодії, тоді як backend забезпечує логіку обробки даних, зберігання інформації та виконання операцій на сервері.

При розробці цих систем часто використовуються технології, такі як HTML, CSS, JavaScript для frontend, й різноманітні серверні мови програмування та бази даних для backend, що дозволяє створити надійну та ефективну платформу для проведення геологічних досліджень.

2.1 Обґрунтування вибору технологій для розробки користувацької частини

Сучасні вимоги до інтерфейсу користувача у галузі геологічної розвідки потребують не лише високої функціональності і зручності використання, але й здатності ефективно взаємодіяти з великими обсягами даних. У цьому контексті, правильний вибір технологій стає вирішальним для досягнення цих цілей.

Обрані технології HTML і CSS забезпечать гнучке та адаптивне оформлення інтерфейсу, JavaScript дозволить реалізувати інтерактивні елементи, а Kendo UI сприятиме швидкому розгортанню стандартних компонентів, які

підвищать продуктивність роботи з даними.

Кожен інструмент відіграє важливу роль у створенні інтуїтивно зрозумілого і візуально привабливого інтерфейсу, що є необхідним для забезпечення високої ефективності використання системи кінцевими споживачами. Розуміння принципів вибору допоможе в подальшому аналізі загальної архітектури інформаційної системи та її впливу на процеси розвідки родовищ.

Для розробки користувацького інтерфейсу інформаційної системи, призначеної для розвідки родовищ корисних копалин, важливо ретельно аналізувати вимоги до зручності та інтуїтивності використання. Ці вимоги є ключовими для забезпечення того, щоб система була не тільки функціональною, але й ефективною і дружньою для кінцевих споживачів. Принципи, які потрібно врахувати при розробці користувацького інтерфейсу:

- Простота доступу до функцій: інтерфейс повинен бути організованим таким чином, щоб основні функції були легко доступні.

- Інтуїтивний дизайн: інтерфейс має відповідати звичним практикам користувача і повинен бути зрозумілим без додаткових інструкцій. Використання стандартних елементів керування (наприклад, кнопок, списків, слайдерів), до яких користувачі звикли, може значно підвищити інтуїтивність системи.

- Доступність та зрозумілість: інтерфейс повинен бути доступним для користувачів з різним рівнем технічної підготовки.

- Швидкість відгуку: система має оперативно реагувати на дії користувача. Затримки у відгуку можуть призвести до фрустрації та помилок, особливо у критичних для безпеки або високонавантажених умовах.

- Адаптивність до різних умов використання: інтерфейс має бути ефективним і зручним для використання в різних умовах – як на великих моніторах, так і на мобільних пристроях. Врахування адаптивності дизайну дозволяє користувачам легше взаємодіяти з системою незалежно від обраного пристрою.

HTML (HyperText Markup Language) відіграє вирішальну роль у структуруванні контенту веб-сторінок і є фундаментом для будь-якого веб-сайту. Це не програмна мова, а мова розмітки, що використовується для створення та форматування розділів на веб-сторінці. HTML відіграє центральну роль у визначенні контенту та його семантичної організації в контексті інтернету [19]. Через свою спроможність структурувати інформацію за допомогою різноманітних тегів, HTML дозволяє розробникам ефективно керувати та представляти дані на веб-сайтах.

Крім того, HTML взаємодіє з CSS (Cascading Style Sheets) для стилізації візуальних елементів і з JavaScript для додання інтерактивності, що робить веб-сторінку більш динамічною й дружньою для користувача [20]. Це може значно підвищити ефективність інформаційних систем розвідки родовищ корисних копалин, надаючи операторам швидкий доступ до даних і підвищуючи точність й швидкість обробки інформації.

Таким чином, HTML є не тільки інструментом для структурування веб-контенту, але й важливим компонентом у створенні ефективних інформаційних систем, які можуть значно впливати на прийняття рішень у георозвідці.

HTML використовується у галузі розвідки родовищ корисних копалин для створення інтерфейсів інформаційних систем, що дозволяють користувачам ефективно взаємодіяти з великими масивами даних (рисуюнок 2.1).



Рисуюнок 2.1. – Інструменти для ІС розвідки родовищ корисних копалин

Інтерактивні карти – HTML використовується для розробки інтерактивних карт, які дозволяють геологам та іншим фахівцям візуально аналізувати

розташування родовищ, їх характеристики та потенційні ризики. Через HTML можна інтегрувати різні інтерактивні елементи, які покращують навігацію та збір даних.

Дашборди (інформаційні панелі) для моніторингу – розробка дашбордів для моніторингу стану родовищ в реальному часі також заснована на HTML. Дашборди забезпечують користувачів важливою інформацією, що включає дані про продуктивність, зміни умов на родовищах та інші критичні метрики, які можуть вимагати швидкого реагування.

Форми для збору даних – HTML використовується для створення форм, які фахівці використовують для збору та введення даних про родовища. Ці форми можуть бути складними і включати поля для вводу числових даних, тексту, вибору дат та інших параметрів, що необхідні для документування та аналізу геологічних даних.

Використання HTML у цих і подібних додатках забезпечує стабільність, доступність і масштабованість інтерфейсів, що є критично важливим для операцій, заснованих на великих масивах даних і вимогах до їх оперативного аналізу.

CSS (Cascading Style Sheets) є фундаментальним інструментом для стилізації веб-сторінок, що дозволяє відділяти вміст веб-сторінки від її візуального представлення [21]. Це розділення критично важливе для ефективного управління веб-ресурсами, оскільки дозволяє розробникам швидко оновлювати та модифікувати вигляд сайту без зміни його структурного коду.

Важливість CSS полягає у його здатності до створення стандартизованих веб-інтерфейсів, які забезпечують універсальність та доступність контенту. Стилiзація, що реалізується через CSS, забезпечує, що веб-сторінки можуть ефективно відображатися на різних пристроях, що має велике значення у сучасному розмаїтті екранних розмірів та резолюцій.

Таким чином, CSS є невід’ємною частиною веб-дизайну, забезпечуючи не тільки зовнішній вигляд сайтів, але й їхню структурну логіку та доступність. Це ключовий елемент, який дозволяє розробникам створювати більш динамічні та

адаптивні веб-досвіди.

Одним з ключових інструментів для забезпечення ефективної взаємодії користувача з інтерфейсом є мова програмування JavaScript, яка дозволяє створювати динамічні та інтерактивні веб-інтерфейси.

JavaScript є багатофункціональною мовою програмування, яка дозволяє реалізовувати складні логічні процеси на стороні клієнта. Основними можливостями JavaScript для покращення взаємодії користувача є:

JavaScript може використовуватись для реалізації наступних функцій у розвідці родовищ корисних копалин:

- Інтерактивні карти – відображення геологічних даних і результатів розвідки на інтерактивних картах з можливістю масштабування, панорамування й перегляду детальної інформації про об'єкти.
- Динамічні звіти – генерація звітів на основі вибраних користувачем параметрів з можливістю інтерактивної взаємодії з даними, наприклад, фільтрація та сортування.
- Онлайн-інструменти аналізу – надання користувачам можливості виконувати складні аналітичні обчислення безпосередньо у веб-браузері, використовуючи JavaScript для обробки даних і відображення результатів.

Таким чином, JavaScript є потужним інструментом для покращення взаємодії користувача з інтерфейсом в автоматизованих інформаційних системах для розвідки родовищ корисних копалин. Використання JavaScript дозволяє створювати динамічні, інтерактивні та зручні у використанні інтерфейси, що сприяє підвищенню ефективності роботи користувачів та забезпечує більш високий рівень якості прийняття рішень на основі аналізу даних.

Kendo UI є потужним фреймворком для розробки користувацьких інтерфейсів, який надає широкий спектр інструментів і віджетів для створення сучасних, функціональних та інтерактивних веб-додатків. Цей фреймворк, розроблений компанією Progress, базується на HTML5 й JavaScript [22], і може використовуватися з різними JavaScript-фреймворками, такими як Angular, React та Vue.js.

Перевагою Kendo UI є підтримка адаптивного дизайну, що дозволяє створювати інтерфейси, які автоматично підлаштовуються під різні розміри екранів і пристрої [23]. Це забезпечує зручність використання додатків, як на настільних комп'ютерах, так і на мобільних пристроях.

У розробці автоматизованих інформаційних систем для розвідки родовищ корисних копалин Kendo UI може бути використано для:

- Відображення даних у вигляді таблиць та графіків: Завдяки потужним компонентам сіток та графіків, Kendo UI дозволяє ефективно візуалізувати геологічні дані та результати розвідки, зокрема, за допомогою таких елементів як сітки, графіки, календарі, випадаючі списки, форми та багато інших. Це дозволяє розробникам швидко створювати складні інтерфейси без потреби в ручній розробці цих компонентів з нуля.

До того ж, віджети Kendo UI легко налаштовуються й розширюються, що дозволяє розробникам створювати унікальні інтерфейси, які відповідають специфічним вимогам їх проєктів. Всі компоненти можна стилізувати за допомогою CSS та налаштовувати їх поведінку за допомогою JavaScript.

- Створення інтерактивних карт. Kendo UI забезпечує інтеграцію з картографічними сервісами, що дозволяє створювати інтерактивні карти для відображення місцезнаходження родовищ та інших важливих об'єктів.

- Розробка користувацьких форм. Kendo UI надає зручні інструменти для створення форм введення даних, які можуть використовуватися для збору та обробки інформації від користувачів. Віджети Kendo UI оптимізовані для забезпечення високої продуктивності, навіть при роботі з великими обсягами даних. Це особливо важливо для бізнес-додатків, де швидкість відображення та обробки даних є критичною вимогою.

В умовах постійно зростаючих загроз кібербезпеки та необхідності захисту конфіденційних даних, правильний вибір технологій актуалізується.

Для досягнення безпеки та надійності HTML-код повинен бути правильно валідований, що зменшить ризики виникнення помилок та вразливостей. Крім того, важливо забезпечити санітизацію введення користувача, що дозволяє

уникнути впровадження шкідливого коду, такого як Cross-Site Scripting (XSS) [24].

Забезпечення безпеки при використанні JavaScript досягається через використання сучасних стандартів програмування, таких як ECMAScript 6+, і уникнення небезпечних практик, таких як `eval()` [25]. Важливо також правильно управляти правами доступу та перевіряти дії користувача, щоб мінімізувати ризики XSS-атак та інших вразливостей.

Використання Kendo UI забезпечує надійність та безпеку через регулярні оновлення, що включають виправлення вразливостей та покращення функціональності. Інтеграція з сучасними веб-технологіями та вбудовані захисні механізми знижують ризик помилок і підвищують загальну стабільність системи.

2.2 Вибір інструментів для розробки серверної частини

Однією з сучасних платформ для розробки веб-додатків є Oracle Application Express (APEX). APEX дозволяє швидко створювати масштабовані, безпечні та зручні у використанні веб-додатки з мінімальними витратами на розробку. Вибір APEX як основи для створення автоматизованої інформаційної системи для розвідки родовищ корисних копалин зумовлений її численними перевагами, серед яких інтеграція з базою даних Oracle, потужні інструменти для розробки інтерфейсу та висока продуктивність.

Oracle APEX працює на основі бази даних Oracle, що забезпечує високу продуктивність і масштабованість. Вбудовані механізми оптимізації запитів і обробки даних дозволяють ефективно працювати з великими обсягами інформації.

APEX також підтримує масштабування додатків, що дозволяє легко адаптувати систему до зростання кількості користувачів і обсягів даних. Це забезпечує стійкість системи до змін у вимогах та умовах експлуатації.

Безпека є ключовим аспектом при розробці інформаційних систем,

особливо тих, що працюють з конфіденційними даними. Oracle APEX надає ряд вбудованих засобів для забезпечення безпеки:

- APEX підтримує різні методи аутентифікації, включаючи вбудовану аутентифікацію, LDAP, SSO та OAuth2. Це дозволяє контролювати доступ користувачів до системи та її ресурсів.
- Вбудовані механізми управління сесіями забезпечують захист від таких атак, як крадіжка сесій та міжсесійні атаки. Використання захищених сеансів гарантує, що дані користувачів залишаються конфіденційними і захищеними.
- APEX має вбудовані засоби для захисту від SQL-ін'єкцій та міжсайтових скриптових атак (XSS). Це досягається через використання параметризованих запитів та автоматичну санітизацію введення користувача [26].

APEX забезпечує високу інтеграцію з іншими продуктами Oracle, такими як Oracle REST Data Services (ORDS), що дозволяє створювати RESTful веб-сервіси для інтеграції з зовнішніми системами, що сприяє створенню комплексних рішень.

Oracle Database є однією з найпотужніших і найнадійніших систем керування базами даних (СКБД), яка використовується у різних галузях, включаючи розвідку родовищ корисних копалин.

Oracle Database відома своєю високою надійністю, що досягається за рахунок таких елементів:

- Відновлення після збоїв: система підтримує механізми резервного копіювання і відновлення даних, що дозволяє мінімізувати втрати даних у разі апаратних або програмних збоїв.
- Реплікація даних: функція реплікації забезпечує синхронізацію даних між різними серверами, що підвищує стійкість системи до збоїв і забезпечує безперервність роботи.
- Кластеризація: технологія Oracle Real Application Clusters (RAC) дозволяє розподіляти навантаження між кількома серверами, забезпечуючи

високу доступність і стійкість до відмов [27].

Oracle Database забезпечує масштабованість як на вертикальному, так і на горизонтальному рівнях:

- Вертикальна масштабованість – збільшення ресурсів (процесорів, пам'яті) на одному сервері для обробки більшого навантаження.
- Горизонтальна масштабованість – додавання нових серверів до кластеру для розподілу навантаження і забезпечення безперервної роботи системи.

Oracle Database надає широкий набір інструментів для управління та аналізу даних (рисунок 2.2).



Рисунок 2.2 – Інструменти Oracle Database для управління та аналізу даних

Ці можливості дозволяють легко адаптувати систему до зростання обсягів даних та кількості користувачів, забезпечуючи її ефективність і продуктивність.

Oracle Database є ключовим компонентом для побудови інформаційної

системи для розвідки родовищ корисних копалин, забезпечуючи високу продуктивність, надійність та масштабованість. Її використання дозволяє ефективно обробляти великі обсяги даних, захищати конфіденційну інформацію та забезпечувати безперервність роботи системи, що є важливим для успішного функціонування інформаційних систем у галузі розвідки корисних копалин.

PL/SQL (Procedural Language/Structured Query Language) – це розширення мови SQL, розроблене компанією Oracle для роботи з базою даних Oracle [28]. PL/SQL дозволяє розробникам створювати складні програми з використанням процедур, функцій, тригерів і пакетів. Використання PL/SQL у розробці автоматизованої інформаційної системи для розвідки родовищ корисних копалин надає значні переваги завдяки високій продуктивності, надійності та широким можливостям програмування.

PL/SQL забезпечує високу продуктивність за рахунок інтеграції з базою даних Oracle і підтримки таких можливостей:

- Збережені процедури та функції – виконання бізнес-логіки безпосередньо на сервері бази даних зменшує обсяг мережевого трафіку та прискорює обробку даних.
- Пакети – групування пов'язаних процедур, функцій і змінних у пакети сприяє організації коду та повторному використанню компонентів, що зменшує час розробки і підтримки додатків.
- Динамічне SQL – можливість виконання SQL-запитів, що формуються динамічно, дозволяє реалізовувати складні запити та оптимізувати їх виконання.

PL/SQL має потужні механізми для забезпечення безпеки:

- Контроль доступу – використання ролей та привілеїв у PL/SQL дозволяє точно контролювати доступ до процедур, функцій і даних. Це забезпечує захист конфіденційної інформації від несанкціонованого доступу.
- Обробка виключень – вбудовані засоби обробки виключень дозволяють належним чином реагувати на помилки і запобігати несанкціонованим діям.

- Безпека програмного коду – використання збережених процедур і функцій знижує ризик SQL-ін'єкцій та інших атак, оскільки код виконується на стороні сервера бази даних.

Надійність PL/SQL досягається завдяки таким механізмам:

- Транзакції – PL/SQL підтримує транзакційне управління, що дозволяє забезпечити атомарність, цілісність, ізоляцію та стійкість (ACID) операцій з даними [29].

- Відновлення після збоїв – PL/SQL надає можливості для відновлення даних після збоїв, забезпечуючи надійність і безперервність роботи системи.

- Модульність коду – розбиття програми на невеликі модулі (процедури, функції, пакети) полегшує тестування та налагодження, що підвищує загальну надійність системи.

PL/SQL є ключовим компонентом у розробці автоматизованої інформаційної системи для розвідки родовищ корисних копалин, надаючи розробникам потужні засоби для створення ефективних, безпечних і надійних програм. Використання PL/SQL дозволяє забезпечити високу продуктивність та масштабованість системи, зберігаючи при цьому контроль над безпекою і цілісністю даних. Це робить PL/SQL незамінним інструментом для створення сучасних інформаційних систем, що відповідають найвищим вимогам до якості та надійності.

2.3 Порівняння обраних інструментів для розробки з аналогами

Розглянемо альтернативні варіанти інструментів для розробки автоматизованої інформаційної системи розвідки родовищ корисних копалин.

HTML і CSS залишаються основними технологіями для розробки веб-інтерфейсів. Вони є основою будь-якого веб-додатку чи сайту, забезпечуючи структуру (HTML) і стиль (CSS) сторінки. Проте, існують деякі альтернативні технології та підходи, такі як Web Components, SVG, Canvas, WebAssembly і No-Code/Low-Code платформи, котрі можуть доповнювати або замінювати HTML

та CSS в залежності від вимог проєкту. Більшість з них мають обмежене застосування або вимагають додаткових знань і навичок.

Серед альтернатив можна відзначити Markdown, який відомий своїм простим синтаксисом і використовується для швидкого написання документації [30]. Однак, через обмежену функціональність він не підходить для створення складних веб-сторінок.

Серед альтернатив JavaScript можна виділити TypeScript, який забезпечує статичну типізацію [31], що полегшує підтримку великих проєктів та інтеграцію з сучасними фреймворками. До обмежень TypeScript можна віднести вимогу компіляції та додаткових знань. Іншим варіантом є CoffeeScript, який спрощує синтаксис JavaScript, роблячи код більш читабельним, але має обмежену спільноту і також потребує компіляції до JavaScript.

Проаналізуємо альтернативи Kendo UI та їх вплив на розробку веб-додатків.

React з бібліотекою Material-UI є однією з популярних альтернатив Kendo UI. React – це бібліотека JavaScript для створення користувацьких інтерфейсів, яка надає можливість побудови компонентів, що забезпечують повторне використання коду та спрощення підтримки великих додатків [32]. Матеріал-UI реалізує принципи Material Design від Google. Основними перевагами цієї комбінації є відповідність принципам Material Design, що надає додаткам сучасний і зручний для користувача вигляд, а також висока гнучкість, яка дозволяє легко налаштовувати компоненти під специфічні потреби проєкту. React має велику і активну спільноту розробників, що забезпечує швидкий доступ до рішень і прикладів.

Водночас, для новачків React може бути складним через вимоги до володіння навичками роботи з JavaScript та розуміння концепцій, таких як компонентний підхід і стан додатку. Крім того, для повноцінного функціонування React-додатку часто потрібні додаткові бібліотеки, такі як Redux для управління станом, що збільшує складність проєкту.

Іншою альтернативою є Angular з Angular Material. Angular – це фреймворк

для розробки веб-додатків, який надає повний набір інструментів для побудови масштабованих та високопродуктивних додатків. Angular Material – це набір компонентів, розроблених з урахуванням принципів Material Design.

Основними перевагами Angular є повнофункціональність, що включає двостороннє зв'язування даних, ін'єкцію залежностей та маршрутизацію, масштабованість, яка ідеально підходить для великих проєктів завдяки своїй архітектурі і суворій типізації через TypeScript. Використання TypeScript забезпечує зниження кількості помилок і полегшує підтримку коду. Проте, Angular має високу криву навчання через свою складність і кількість концепцій [33], які необхідно освоїти, що робить його складним для новачків.

Vue.js з Vuetify є ще однією популярною альтернативою. Vue.js – це прогресивний фреймворк для створення користувацьких інтерфейсів [34], відомий своєю простотою і гнучкістю. Основними перевагами Vue.js є простий і зрозумілий синтаксис, що робить його легким у вивченні і використанні, а також гнучкість, яка дозволяє створювати додатки різної складності, від простих компонентів до складних односторінкових додатків. Vue.js відомий своєю низькою кривою навчання, що робить його привабливим для новачків. Водночас, Vue.js має меншу спільноту розробників у порівнянні з React або Angular, що може ускладнити пошук рішень для специфічних проблем. Для дуже великих проєктів Vue.js може бути менш придатним у порівнянні з Angular, що спеціалізується на масштабованості.

У порівнянні з іншими популярними технологіями, такими як React, Angular, Vue.js та Bootstrap, обрані технології мають кілька переваг. HTML, CSS та JavaScript є загальноприйнятими стандартами, що забезпечують високу сумісність і підтримку у всіх сучасних браузерах. Це робить їх надійним вибором для довгострокових проєктів. Kendo UI, завдяки комерційній підтримці та детальній документації, дозволяє розробникам швидко вирішувати складні завдання та забезпечувати високу якість інтерфейсів.

Для оцінки обраних фронтенд технологій у таблиці 2.1, проведемо порівняння з іншими поширеними технологіями. Це дозволить нам не тільки

зрозуміти переваги та недоліки обраних рішень, але й побачити їх у контексті сучасних тенденцій та вимог

Таблиця 2.1 – Порівняльна таблиця фронтенд технологій

Показник и	Продуктив - ність	Гнучкість	Підтримка	Легкість використання	Інтеграція
HTML	Висока, основа для веб-сторінок	Середня, обмежена можливостями мови розмітки	Висока, загальноприйнятий стандарт	Висока, базовий рівень	Висока, основа для будь-яких фреймворків
CSS	Висока, ефективна стилізація	Висока, гнучкість через препроцесори (Sass, LESS)	Висока, загальноприйнятий стандарт	Висока, легко навчатися основам	Висока, використовуються у всіх фреймворках
JavaScript	Висока, динамічний контент	Висока, через бібліотеки та фреймворки	Висока, активна спільнота	Середня, складні концепції для новачків	Висока, підтримка у всіх браузерах
Kendo UI	Висока, оптимізована - ні компоненти	Висока, багато налаштувань	Висока, комерційна підтримка	Середня, складні для налаштування компоненти	Висока, інтеграція з іншими бібліотеками JavaScript
React	Висока, віртуальний DOM	Висока, компоненти та хуки	Висока, підтримка від Facebook	Середня, потребує знань JavaScript	Висока, багато інструментів для інтеграції
Angular	Висока, залежить від архітектур	Висока, потужний CLI та модулі	Висока, підтримка від Google	Середня, складна структура	Висока, інтеграція з багатьма сервісами Google
Vue.js	Висока, реактивність	Висока, проста інтеграція з існуючими проєктами	Висока, активна спільнота	Висока, проста у використанні	Висока, інтеграція з іншими бібліотеками
Bootstrap	Висока, компоненти та сітка	Висока, кастомізація через Sass	Висока, підтримка від Twitter	Висока, легкий старт	Висока, інтеграція з усіма фреймворками

Порівняємо Oracle Application Express (APEX), Oracle Database, PL/SQL з їх аналогами, такими як Microsoft Power Apps, SQL Server і Transact-SQL (T-SQL), а також з іншими поширеними рішеннями для створення інформаційних систем.

Oracle Application Express (APEX) є потужною платформою для швидкої розробки веб-додатків, інтегрованою з Oracle Database, що забезпечує високу продуктивність та надійність. Аналогічно, Microsoft Power Apps також пропонує швидку розробку бізнес-додатків, але більше орієнтований на інтеграцію з Microsoft Azure та іншими сервісами Microsoft. Обидві платформи забезпечують високу безпеку, але Oracle APEX має перевагу, завдяки глибокій інтеграції з Oracle Database та її потужними засобами управління даними.

Oracle Database відзначається своєю продуктивністю, масштабованістю і надійністю, що робить її ідеальною для роботи з великими обсягами даних. Вона забезпечує паралельну обробку запитів, індексацію та кешування, що значно підвищує швидкість доступу до даних. У порівнянні з SQL Server, який також є потужною СКБД, Oracle Database надає більш широкі можливості для налаштування продуктивності та управління даними. SQL Server, хоча і забезпечує високу продуктивність, краще підходить для інтеграції з продуктами Microsoft, такими як Power BI та Azure.

PL/SQL, як розширення мови SQL, надає розробникам потужні можливості для створення процедур, функцій та тригерів, що дозволяє виконувати складні бізнес-процеси безпосередньо на сервері бази даних. Це значно зменшує обсяг мережевого трафіку і покращує продуктивність додатків. У порівнянні з Transact-SQL (T-SQL) від Microsoft, який використовується в SQL Server, PL/SQL пропонує подібні можливості для програмування, але краще інтегрований з функціоналом Oracle Database. Обидві мови забезпечують високу безпеку і підтримують транзакційне управління, але PL/SQL часто вважається більш гнучким та потужним інструментом для складних корпоративних додатків.

Кожна з цих технологій має свої переваги та особливості, що можуть

вплинути на продуктивність, безпеку, надійність, масштабованість та інтеграцію системи. Нижче наведена таблиця 2.2, яка порівнює ці технології за ключовими критеріями, що допоможе зробити обґрунтований вибір для конкретних вимог проєкту.

Таблиця 2.2 – Порівняльна таблиця бекенд технологій

Показники	Продуктивність	Безпека	Надійність	Масштабованість
Oracle APEX	Висока, завдяки інтеграції з Oracle Database	Висока, контроль доступу, шифрування даних	Висока, з резервним копіюванням та відновленням	Висока, масштабування на рівні бази даних
Microsoft Power Apps	Висока, орієнтована на швидку розробку бізнес-додатків	Висока, інтеграція з Azure Active Directory	Висока, за рахунок підтримки хмарних технологій	Висока, інтеграція з Microsoft Azure
Oracle Database	Дуже висока, паралельна обробка, індексація, кешування	Висока, аутентифікація, шифрування, аудит	Дуже висока, підтримка кластеризації та відновлення	Дуже висока, горизонтальна і вертикальна масштабованість
SQL Server	Висока, ефективна обробка запитів	Висока, підтримка Active Directory, шифрування	Висока, підтримка реплікації та резервного копіювання	Висока, інтеграція з Azure для масштабування
PL/SQL	Висока, збережені процедури та функції	Висока, контроль доступу, обробка виключень	Висока, підтримка транзакцій	Висока, підтримка паралельної обробки
Transact-SQL (T-SQL)	Висока, ефективна обробка процедур	Висока, підтримка ролей та привілеїв	Висока, підтримка транзакцій	Висока, підтримка паралельної обробки

Microsoft Power Apps, SQL Server та Transact-SQL (T-SQL) мають свої переваги, такі як інтеграція з іншими продуктами Microsoft, швидка розробка бізнес-додатків, але вони поступаються вибраним технологіям Oracle за показниками продуктивності, інтеграції та підтримки великих обсягів даних.

2.4 Висновки до розділу 2

Використання Kendo UI дозволяє значно прискорити розробку завдяки готовим компонентам і багатофункціональним віджетам. Альтернативи, такі як React з Material-UI, Angular з Angular Material та Vue.js з Vuetify, також пропонують потужні інструменти для створення сучасних інтерфейсів, але можуть вимагати більше часу на навчання та інтеграцію.

З огляду на поставлене технічне завдання, вибір HTML, CSS, JavaScript та Kendo UI, PL/SQL, Oracle APEX та Oracle Database є оптимальним рішенням, оскільки ці технології забезпечують баланс між простотою використання, потужністю та ефективністю. Вони дозволяють швидко створювати функціональні та привабливі веб-додатки, що відповідають сучасним стандартам веб-розробки.

Вибір Oracle APEX, Oracle Database та PL/SQL є найбільш оптимальним для досягнення високих результатів у розробці та впровадженні автоматизованої інформаційної системи для розвідки родовищ корисних копалин. Ці технології забезпечать стабільну роботу системи, її масштабованість і безпеку, що є критичними для успішного завершення проєкту.

Правильне використання вказаних технологій забезпечує захист даних, стійкість до зовнішніх загроз і стабільність роботи додатка, які є ключовими елементами успішного функціонування інформаційної системи.

3 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ, ЇЇ ТЕХНОЛОГІЧНИХ МОДУЛІВ ТА КОМПОНЕНТІВ

Проектування архітектури інформаційної системи є ключовим етапом у створенні ефективного та надійного програмного забезпечення. У цьому розділі ми розглянемо основні аспекти проектування інформаційної системи для розвідки родовищ корисних копалин, зокрема архітектуру системи та впровадження її технологічних модулів, таких як фронтенд, бекенд, бази даних і інтеграційні компоненти.

Фронтенд є обличчям системи, і його розробка потребує уваги до деталей для забезпечення інтуїтивності та зручності користувачів. Основні кроки проектування фронтенду включають розробку макетів, створення HTML-структури, стилізацію з CSS, додавання динаміки з JavaScript, а також інтеграцію Kendo UI.

Бекенд обробляє всю бізнес-логіку та забезпечує збереження і обробку даних. Основні кроки включають вибір технологій, створення API, реалізацію бізнес-логіки, тестування та оптимізацію.

База даних є центральним елементом системи, де зберігаються всі дані. Основні кроки проектування включають: створення схеми бази даних, налаштування безпеки, реалізацію індексів та ключів, а також оптимізацію запитів.

Інтеграційні компоненти забезпечують взаємодію між різними частинами системи та зовнішніми сервісами. Основні аспекти включають розробку налаштування системи обміну даними, а також моніторинг та аудит змін.

Проектування інформаційної системи для розвідки родовищ корисних копалин є складним процесом, що включає визначення архітектури системи та впровадження її технологічних модулів. Вибір технологій, таких як HTML, CSS, JavaScript, Kendo UI, PL/SQL, та Oracle Database, забезпечує високу продуктивність, надійність, безпеку та масштабованість системи. Ретельне проектування та впровадження кожного з компонентів гарантують створення

ефективної та надійної інформаційної системи, яка здатна реалізовувати визначені завдання з розвідки родовищ корисних копалин.

3.1 Архітектура інформаційної системи

Архітектуру інформаційної системи можна описати як концепцію, що визначає модель, структуру, виконувані функції й взаємозв'язок компонентів інформаційної системи [35]. Для системи розвідки родовищ корисних копалин важливо обрати архітектуру, що забезпечує масштабованість, продуктивність, безпеку і легкість у підтримці. Інформаційна система складається з декількох ключових компонентів: фронтенд, бекенд, база даних та інтеграційні компоненти (рисунок 3.1).

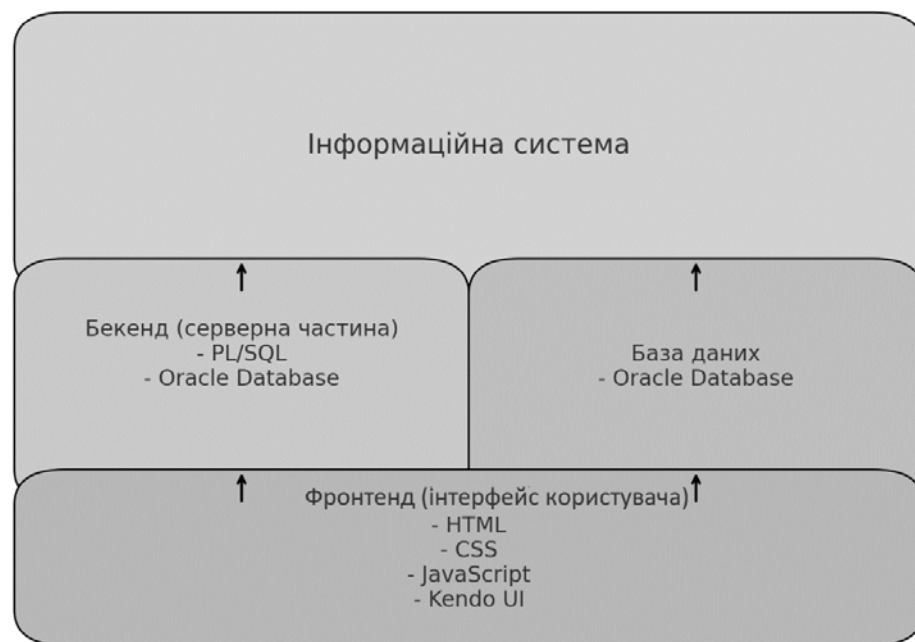


Рисунок 3.1. – Архітектура інформаційної системи для розвідки родовищ корисних копалин

Фронтенд відповідає за взаємодію з користувачем і реалізований з використанням: HTML для структури сторінок, CSS для стилізації, JavaScript для динамічної взаємодії та Kendo UI для інтеграції готових компонентів.

Бекенд обробляє бізнес-логіку та взаємодію з базою даних,

використовуючи PL/SQL та Oracle Database для виконання запитів і збереження даних.

На рисунку 3.2 зображено діаграму послідовностей автоматизованої інформаційної системи для розвідки родовищ корисних копалин

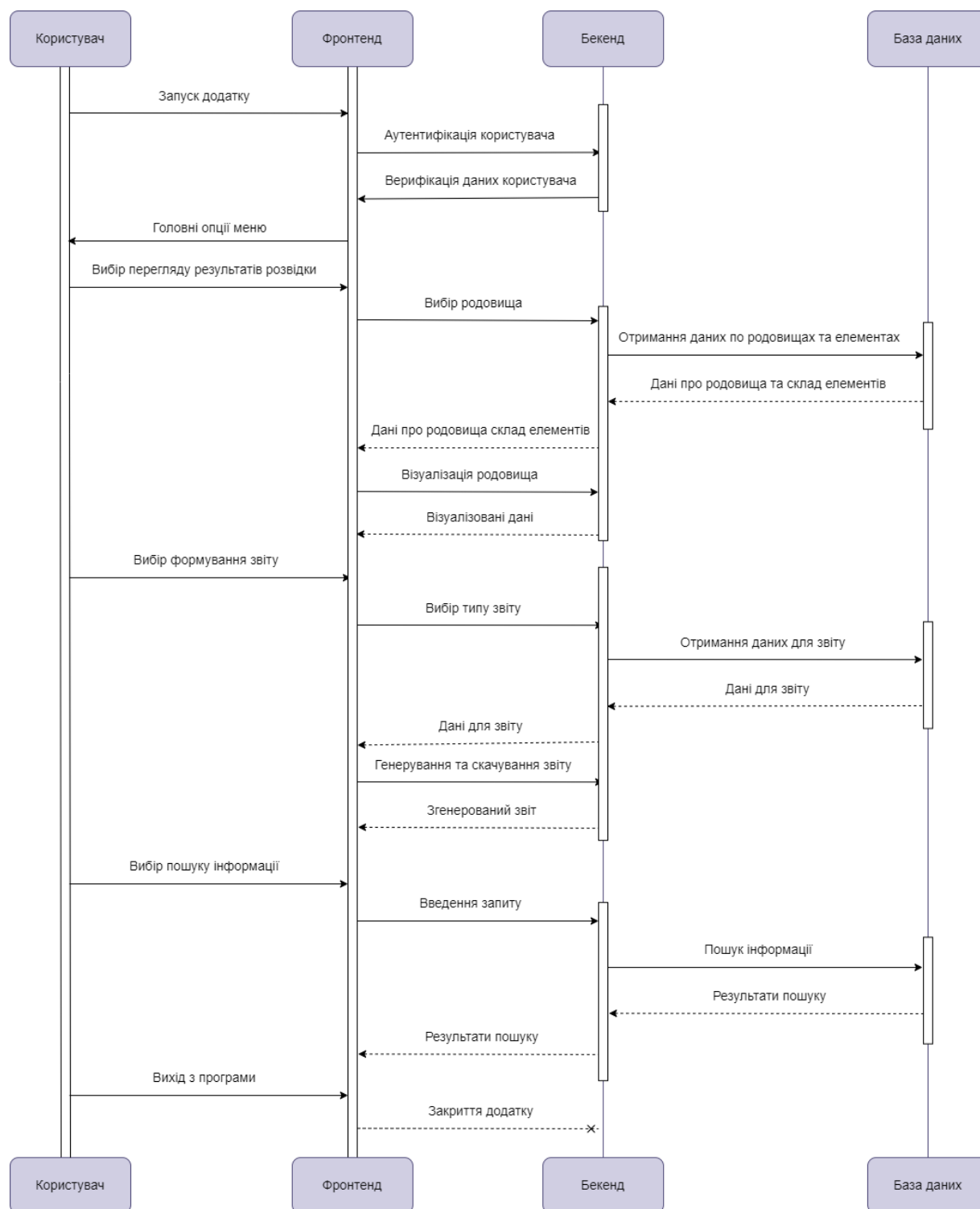


Рисунок 3.2 –Діаграма послідовностей автоматизованої інформаційної системи для розвідки родовищ корисних копалин

Діаграма послідовностей дозволяє детально розглянути взаємодію між різними модулями системи та визначити критичні точки, що впливають на ефективність процесу розвідки. Це сприяє оптимізації роботи системи та підвищенню точності прогнозів щодо родовищ корисних копалин.

Інтеграційні компоненти забезпечують взаємодію між різними частинами системи, включаючи розробку API, налаштування систем обміну даними, моніторинг і логування.

Обрана архітектура забезпечує ефективність, надійність і масштабованість інформаційної системи, дозволяючи їй виконувати всі необхідні завдання з розвідки родовищ корисних копалин.

3.2 Аналіз функціональних вимог користувача

Аналіз функціональних вимог користувача є критично важливим етапом розробки будь-якої інформаційної системи. Він дозволяє визначити, які саме функції і можливості повинна забезпечувати система для задоволення потреб користувачів. Функціональні вимоги визначають, що система повинна робити, які задачі виконувати і які послуги надавати.

На початковому етапі проєкту проводиться збір вимог користувачів. Важливо враховувати всі категорії користувачів, що будуть взаємодіяти з системою: кінцевих користувачів, адміністраторів, технічний персонал тощо.

Однією з ключових складових будь-якої інформаційної системи є забезпечення зручного та інтуїтивно зрозумілого інтерфейсу для введення даних.

Від правильно спроектованого інтерфейсу залежить ефективність роботи системи, задоволеність користувачів та загальна продуктивність (рисунок 3.3). Користувачі повинні мати можливість швидко та безпомилково вводити необхідну інформацію, не витрачаючи зайвий час на вивчення складних інтерфейсів чи некоректне введення даних [36]. Це особливо важливо для систем, що працюють з великим обсягом даних, або де точність введення інформації є критичною.



Рисунок 3.3 – Принципи розробки інтерфейсу для введення даних

Забезпечення зручного та інтуїтивно зрозумілого інтерфейсу для введення даних передбачає кілька важливих аспектів:

1. Інтерфейс повинен бути максимально простим та зрозумілим для користувача. Це означає, що форми введення даних мають бути логічно структурованими, з чіткими підказками та поясненнями для кожного поля. Використання стандартних елементів управління, таких як текстові поля, випадаючі списки, радіокнопки та чекбокси, сприяє швидкому освоєнню інтерфейсу новими користувачами [37].

2. Для запобігання введенню некоректної або неповної інформації інтерфейс повинен забезпечувати валідацію даних в реальному часі. Це включає перевірку формату введених даних, обов'язковість заповнення певних полів, перевірку допустимих значень та інші правила. Відповідні повідомлення про помилки та підказки повинні допомагати користувачеві виправити введену інформацію.

3. Використання механізмів автозаповнення та підказок значно спрощує процес введення даних. Наприклад, при введенні адреси або імені система може пропонувати варіанти завершення, базуючись на попередньо введених даних або загальних шаблонах. Це не тільки прискорює введення, але й знижує ймовірність помилок.

4. Інтерфейс повинен бути адаптивним, тобто зберігати свою функціональність та зручність на різних пристроях, таких як комп'ютери, планшети та мобільні телефони. Це дозволяє користувачам працювати з системою у будь-яких умовах, не втрачаючи в зручності та ефективності.

Запровадження цих принципів при розробці інтерфейсу для введення даних забезпечить його зручність, інтуїтивність та ефективність, що в свою чергу сприятиме підвищенню загальної продуктивності та задоволеності користувачів інформаційною системою.

Після введення даних система повинна забезпечувати їх ефективну обробку, яка включає:

- Розрахунки – виконання складних математичних та статистичних розрахунків на основі введених даних. Це може включати обчислення середніх значень, медіан, стандартних відхилень та інших статистичних показників.
- Аналіз даних – виконання аналізу даних для виявлення трендів, аномалій та кореляцій. Використання аналітичних інструментів для прогнозування та ухвалення рішень.
- Обробка даних – фільтрація, сортування та агрегація даних для підготовки їх до подальшого використання. Можливість комбінування даних з різних джерел та їх нормалізація.

Ефективна обробка даних є критично важливою для будь-якої інформаційної системи, особливо для автоматизованих систем розвідки родовищ корисних копалин. Вона забезпечує високу точність обчислень та аналізу, що є основою для ухвалення правильних управлінських рішень щодо оцінки та розробки родовищ. Точні дані дозволяють геологам та інженерам приймати обґрунтовані рішення, які можуть вплинути на подальший розвиток геологорозвідувальних проєктів.

Швидка обробка даних дає можливість оперативно реагувати на зміни у геологічній інформації або в умовах експлуатації родовищ.

Автоматизація обробки геологічних даних значно знижує необхідність у ручній праці, що дозволяє скоротити час та зусилля, витрачені на виконання

рутинних завдань, таких як аналіз зразків чи моделювання родовищ. Це також знижує ризик людських помилок, що підвищує загальну надійність і точність системи.

Ефективна обробка даних сприяє виконанню глибокого аналізу та прогнозування щодо перспективності родовищ. Це дає можливість передбачати ринкові тенденції та ефективно планувати видобуток корисних копалин, забезпечуючи конкурентні переваги для гірничодобувних компаній.

Таким чином, ефективна система обробки даних розвідки забезпечує користувачів необхідною інформацією, котрі стають основою для прийняття обґрунтованих рішень. Впровадження цих принципів дозволяє інформаційній системі не тільки зберігати дані, але й максимально ефективно використовувати їх для аналітики та управління рішеннями і дозволяє реалізувати повний потенціал системи.

На рисунку 3.4 наведено основні засоби виведення даних, які повинна забезпечувати система.



Рисунок 3.4 – Засоби виведення даних автоматизованої інформаційної системи для розвідки родовищ корисних копалин

Ефективність автоматизованої інформаційної системи для розвідки родовищ корисних копалин також залежить від здатності надавати зручні та

інформативні засоби для виведення даних. Це дозволяє користувачам, таким як геологи, інженери та менеджери, швидко та ефективно отримувати необхідну інформацію для ухвалення обґрунтованих рішень. Розглянемо основні засоби виведення даних, які повинна забезпечувати система.

Одним з ключових елементів цієї системи є інтерфейс «Полігони», який представляє собою місця зберігання та обробки матеріалів. Цей елемент дозволяє організовувати і структурувати дані про матеріальні ресурси, що використовуються або видобуваються в процесі розвідки, а також забезпечує можливість відстеження і контролю всіх пов'язаних з цим процесів.

На рисунку 3.5 представлено варіанти використання функціонального елемента «Полігони» в інформаційній системі «Розвідка родовищ» у вигляді діаграми.

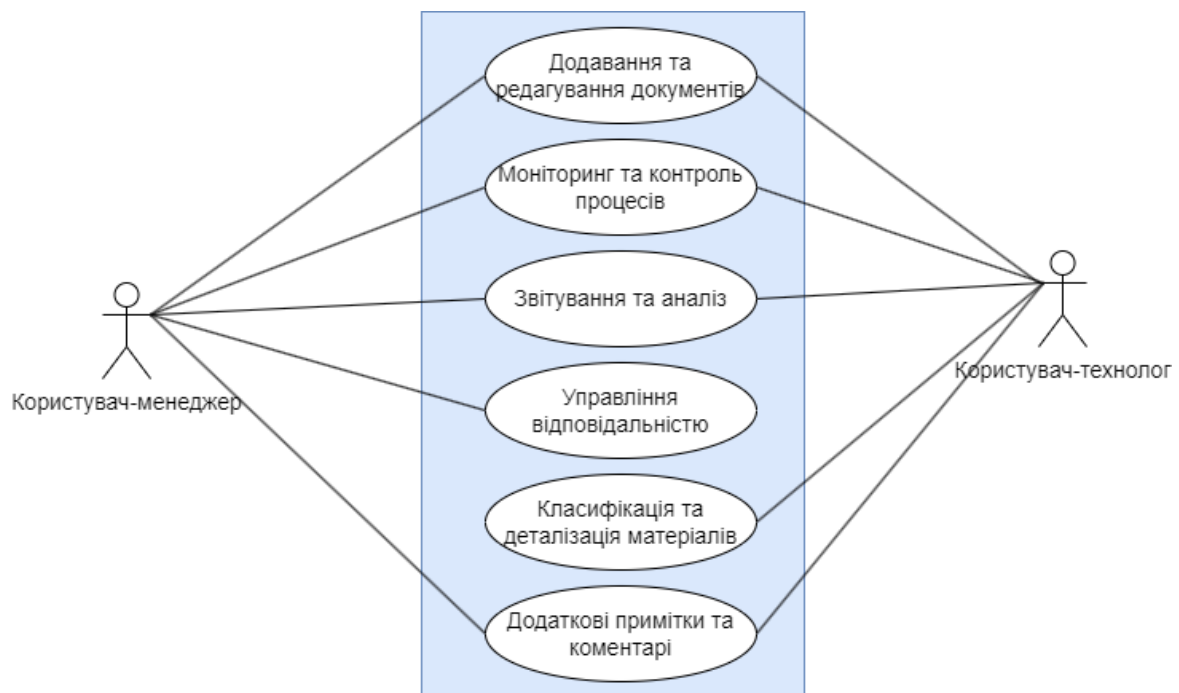


Рисунок 3.5 – Діаграма варіантів використання функціонального елемента «Полігони»

Основний користувач має можливість взаємодії з системою через наступні функції:

— Організація місць зберігання — користувач може точно ідентифікувати та організовувати місця зберігання матеріалів.

- Моніторинг та контроль процесів – відстеження хронології подій, виконавців і типів документів, що супроводжують матеріали.
- Звітування та аналіз – зібрана інформація використовується для створення звітів та аналізу ефективності робіт, що допомагає оптимізувати процеси та приймати обґрунтовані рішення.
- Управління відповідальністю – призначення виконавців та відстеження їхньої роботи дозволяє чітко розподілити відповідальність і контролювати виконання завдань.
- Класифікація та деталізація матеріалів – поділ матеріалів на сегменти, сектори та інші категорії допомагає структурувати дані, полегшуючи їх пошук та аналіз.
- Додаткові примітки та коментарі – зберігання додаткової інформації про матеріали та процеси.

Діаграма послідовностей (рисунок 3.6) відображає взаємодію користувачів та компоненту «Документи» автоматизованої інформаційної системи для розвідки родовищ корисних копалин.



Рисунок 3.6 – Діаграма варіантів використання функціонального елемента «Документи»

Розглянемо кожний з варіантів використання компоненту «Документи»:

- Введення та редагування загальної інформації – цей процес включає введення нових даних в систему та редагування вже існуючої інформації.
- Перегляд та управління пов'язаними документами – користувачі можуть переглядати, додавати, змінювати та видаляти документи, що пов'язані з розвідкою родовищ.
- Комунікація через чат – функція, що дозволяє користувачам спілкуватися між собою для координації дій та обміну інформацією.
- Налаштування параметрів запису – цей процес включає встановлення параметрів для збереження даних у системі.
- Управління файлами – функція, що дозволяє користувачам організовувати та керувати файлами, що зберігаються в системі.
- Введення та аналіз даних проб – користувачі вводять дані про проби та проводять їх аналіз для визначення перспективності родовищ.

На діаграмах присутні два типи користувачів: користувач-менеджер і користувач-технолог.

Користувач-менеджер виконує більшість зазначених функцій, ініціюючи процеси введення, редагування, перегляду, управління документами і файлами, а також комунікації та аналізу даних.

Користувач-технолог переважно взаємодіє з функціями перегляду документів, комунікації, налаштування параметрів запису, управління файлами та введення й аналізу даних.

Взаємодія між користувачами та функціями системи забезпечує комплексний підхід до розвідки родовищ корисних копалин, що сприяє оптимізації процесу та підвищенню ефективності роботи системи.

3.3 Проектування бази даних

Проектування бази даних є важливим етапом розробки автоматизованої інформаційної системи для розвідки родовищ корисних копалин. Від якості

проектування бази даних залежить ефективність зберігання, обробки та виведення даних, а також загальна продуктивність системи.

Аналіз UML-діаграм показав, що розробка цього модуля програмного забезпечення потребує інтеграції з базою даних. На початковому етапі розробки баз даних важливо оцінити доцільність використання певної моделі, зокрема ієрархічної, мережевої, реляційної чи об'єктно-орієнтованої.

Ієрархічні та мережеві моделі не відповідають сучасним вимогам. Ієрархічна модель має тенденцію до надлишковості даних при складніших структурах, тоді як мережева модель ускладнює обробку запитів [38]. Натомість, сучасні реляційні та нереляційні бази даних пропонують широкий вибір рішень. Вибір конкретної моделі базується на специфічних задачах, що виникають під час розробки.

Для системи, що проектується, було обрано реляційну модель бази даних, що зумовлено її перевагами. По-перше, реляційна модель дозволяє ефективно зберігати та обробляти великі обсяги даних, що є критичним для системи розвідки родовищ корисних копалин. По-друге, вона забезпечує високу швидкість доступу до даних та підтримує складні запити, що дозволяє оперативно отримувати необхідну інформацію. Крім того, реляційні бази даних відомі своєю надійністю й можливістю забезпечення цілісності даних, що є важливим для збереження точності та достовірності інформації.

Важливо враховувати, як переваги та недоліки обраної моделі відобразяться у конкретній предметній області. У випадку системи для розвідки родовищ корисних копалин необхідно звернути увагу на ефективність обробки великих обсягів даних, швидкість доступу до них та підтримку складних запитів.

Розглянемо основні аспекти проектування бази даних, включаючи визначення вимог, моделювання даних та забезпечення цілісності даних.

Визначення вимог є першим кроком у цьому процесі. Це включає збір інформації про типи даних, які будуть зберігатися, обсяги даних, частоту їх оновлення, вимоги до доступності та безпеки. Визначення вимог допомагає сформулювати чітке уявлення про структуру даних і забезпечити ефективне їх

зберігання та обробку.

Перше, що потрібно визначити, це типи даних, які будуть зберігатися в базі даних. Для системи розвідки родовищ корисних копалин це будуть:

- Текстові дані (опис родовищ, коментарі фахівців).
- Числові дані (вимірювання концентрації хімічних елементів, координати родовищ).
- Дати (дата збору зразків, дата перевірки зразків).
- Файли (протоколи вимірювання, звіти по зібраних зразках).

Частота оновлення даних є важливим параметром для налаштування ефективної роботи бази даних у будь-якій інформаційній системі. У системі розвідки родовищ корисних копалин планується оновлення даних щоденно, по декілька разів на день, з можливістю збереження. Це важливий аспект, котрий потрібно враховувати при проєктуванні й налаштуванні бази даних, щоб забезпечити стабільну роботу та оперативний доступ до актуальної інформації. Щоб ефективно обробляти щоденні оновлення, система повинна мати налаштовані механізми автоматизації процесів введення і перевірки даних, що допоможе знизити ймовірність помилок й підвищити продуктивність роботи.

Оновлення даних по декілька разів на день передбачає частіше внесення нових даних або змін до існуючих записів. Це особливо важливо в ситуаціях, коли дані мають оперативний характер і потребують швидкої реакції. Наприклад, під час активних досліджень або аналізів нові дані можуть надходити кілька разів на день. Для забезпечення безперебійної роботи системи важливо налаштувати механізми обробки пакетних оновлень та інкрементальних змін, що дозволить уникнути надмірного навантаження на базу даних і зберегти високу швидкість доступу до інформації [39].

Основна ціль даної розробки – опрацювання даних про розвідку родовищ корисних копалин та їх аналіз. Це включає інформацію про кількість і види корисних копалин, методи дослідження, матеріальні цінності, загальну інформацію про виконані розвідки, яка на даний момент обробляється на обраному підприємстві, та деталі про знайдені ресурси.

З урахуванням технічного завдання спроектовано структуру бази даних (БД) системи «Розвідка родовищ», що включає в себе 5 логічних сутностей: документація, типи документів, матеріальні цінності, хімічні елементи, локації, спеціалісти. Відповідно до цих п'яти сутностей було створено п'ять таблиць, які представлені у структурі БД з відповідними зв'язками (рисунок 3.7).

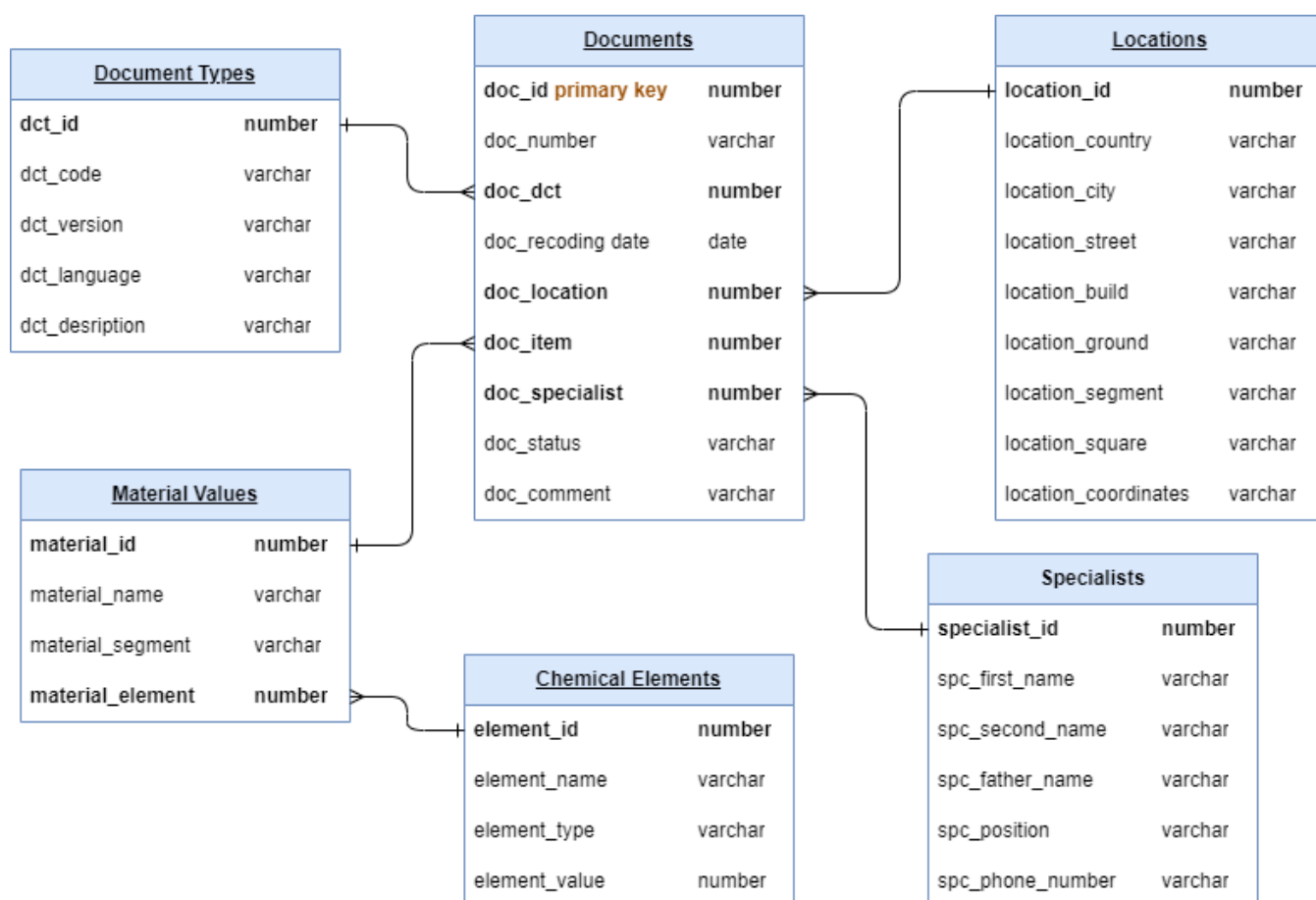


Рисунок 3.7 – Структура бази даних для інформаційної системи розвідки родовищ корисних копалин

Для проектування бази даних системи «Розвідка родовищ» були використані різні типи змінних для забезпечення коректного зберігання та обробки даних. Нижче наведено опис типів змінних, що використовуються в полях таблиць бази даних.

Тип змінної *number* використовується для полів, що зберігають числові значення. Це можуть бути ідентифікатори (ID), значення хімічних елементів та інші числові дані. Поля цього типу забезпечують цілісність і унікальність даних,

особливо коли вони використовуються як первинні ключі.

Тип змінної `varchar` використовується для полів, що зберігають текстові дані змінної довжини. Цей тип є гнучким для зберігання рядків різної довжини, наприклад, назв, кодів, описів, імен, адрес та коментарів.

Тип змінної `date` використовується для полів, що зберігають календарні дати. Поля цього типу дозволяють зберігати інформацію про дати у стандартному форматі, що полегшує сортування, фільтрацію та порівняння дат.

Первинний ключ (`primary key`) використовується для унікальної ідентифікації кожного запису в таблиці. Поля, визначені як первинні ключі, повинні містити унікальні значення і не можуть бути порожніми. В даній базі даних первинні ключі зазвичай мають тип `number`.

Зовнішні ключі (`foreign key`) використовуються для встановлення зв'язків між таблицями. Вони вказують на первинний ключ іншої таблиці і забезпечують цілісність даних між зв'язаними таблицями. Наприклад, поле `doc_dct` в таблиці `Documents` є зовнішнім ключем, що посиляється на поле `dct_id` в таблиці `Document Types`.

Таблиця `Documents` є центральною частиною бази даних системи «Розвідка родовищ». Вона містить детальну інформацію про документи, що використовуються в процесі розвідки родовищ корисних копалин. Нижче наведено максимально опис полів цієї таблиці:

`doc_id` (`number`, `primary key`): Унікальний ідентифікатор документа. Використовується для унікальної ідентифікації кожного документа в таблиці. Це поле є первинним ключем, що гарантує унікальність кожного запису.

`doc_number` (`varchar`): Номер документа. Використовується для зберігання номера документа, який може бути корисним для пошуку та ідентифікації документа. Тип даних – текстове поле змінної довжини (`varchar`).

`doc_dct` (`number`, `foreign key`): Ідентифікатор типу документа. Використовується для визначення типу документа шляхом посилання на поле `dct_id` в таблиці `Document Types`. Тип даних - ціле число (`number`). Є зовнішнім ключем, що забезпечує зв'язок із таблицею `Document Types`.

`doc_recoding_date` (date): Дата реєстрації документа. Зберігає дату, коли документ був зареєстрований або створений в системі.

`doc_location` (number, foreign key): Ідентифікатор локації. Використовується для вказівки на місце розташування, пов'язане з документом, шляхом посилання на поле `location_id` в таблиці `Locations`. Є зовнішнім ключем, що забезпечує зв'язок із таблицею `Locations`.

`doc_item` (number, foreign key): Ідентифікатор матеріалу. Використовується для вказівки на матеріальну цінність, пов'язану з документом, шляхом посилання на поле `material_id` в таблиці `Material Values`. Є зовнішнім ключем, що забезпечує зв'язок із таблицею `Material Values`.

`doc_specialist` (number, foreign key): Ідентифікатор спеціаліста. Використовується для вказівки на спеціаліста, пов'язаного з документом, шляхом посилання на поле `specialist_id` в таблиці `Specialists`. Є зовнішнім ключем, що забезпечує зв'язок із таблицею `Specialists`.

`doc_status` (varchar): Статус документа. Зберігає поточний статус документа, що може відображати етап обробки або актуальність документа.

`doc_comment` (varchar): Коментар до документа. Використовується для зберігання додаткових приміток або коментарів щодо документа.

Таблиця `Document Types` є важливою частиною бази даних системи «Розвідка родовищ». Вона містить інформацію про різні типи документів, що використовуються в системі. Нижче розглянемо кожен з полів цієї таблиці.

`dct_id` (number, primary key): Унікальний ідентифікатор типу документа. Використовується для унікальної ідентифікації кожного типу документа в таблиці. Це поле є первинним ключем, що гарантує унікальність кожного запису.

`dct_code` (varchar): Код типу документа. Використовується для зберігання коду, що ідентифікує тип документа.

`dct_version` (varchar): Версія типу документа. Використовується для вказівки версії документа, що може бути корисним для відстеження змін та модифікацій документа.

`dct_language` (varchar): Мова документа. Зберігає інформацію про мову, на

якій написаний документ.

dct_description (varchar): Опис типу документа. Використовується для зберігання детального опису або приміток про тип документа.

Таблиця Document Types пов'язана з таблицею Documents через зовнішній ключ doc_dct в таблиці Documents, який посилається на поле dct_id в таблиці Document Types. Це забезпечує зв'язок між документами та їх типами, дозволяючи системі відстежувати, які типи документів використовуються для кожного конкретного випадку.

Таблиця Locations призначена для зберігання та управління інформацією про географічні місця розташування об'єктів розвідки в системі «Розвідка родовищ». Вона містить дані про країни, міста, вулиці, будівлі, земельні ділянки, сегменти, площі та координати локацій. Це дозволяє системі ефективно організовувати та відстежувати місця розвідки, полегшуючи їх пошук та використання. Таблиця Locations містить поля:

location_id (number, primary key): Унікальний ідентифікатор локації. Використовується для унікальної ідентифікації кожного місця розташування в таблиці. Це поле є первинним ключем, що гарантує унікальність кожного запису.

location_country (varchar): Країна. Зберігає назву країни, де розташована локація.

location_city (varchar): Місто. Зберігає назву міста, де знаходиться локація.

location_street (varchar): Вулиця. Зберігає назву вулиці, на якій розташована локація.

location_build (varchar): Будівля. Зберігає інформацію про будівлю, де розташована локація.

location_ground (varchar): Земельна ділянка. Зберігає інформацію про земельну ділянку, на якій розташована локація.

location_segment (varchar): Сегмент локації. Зберігає інформацію про сегмент або частину локації.

location_square (varchar): Площа локації. Зберігає інформацію про площу локації.

`location_coordinates` (varchar): Координати локації. Зберігає географічні координати (широта і довгота) локації.

Таблиця `Locations` пов'язана з таблицею `Documents` через зовнішній ключ `doc_location` в таблиці `Documents`, який посилається на поле `location_id` в таблиці `Locations`. Це забезпечує зв'язок між документами та місцями розташування, дозволяючи системі відстежувати, які документи пов'язані з кожною конкретною локацією.

Таблиця `Specialists` є важливою для зберігання та управління інформацією про спеціалістів, залучених до процесу розвідки родовищ у системі «Розвідка родовищ». Вона містить дані про імена, прізвища, по батькові, посади та номери телефонів спеціалістів. Це дозволяє системі ефективно організовувати та відстежувати інформацію про фахівців, полегшуючи їх пошук та використання. Розглянемо поля даної таблиці:

`specialist_id` (number, primary key): Унікальний ідентифікатор спеціаліста. Використовується для унікальної ідентифікації кожного спеціаліста в таблиці. Це поле є первинним ключем, що гарантує унікальність кожного запису. Тип даних - ціле число (number).

`sps_first_name` (varchar): Ім'я спеціаліста. Зберігає ім'я спеціаліста.

`sps_second_name` (varchar): Прізвище спеціаліста. Зберігає прізвище спеціаліста.

`sps_father_name` (varchar): По батькові спеціаліста. Зберігає по батькові спеціаліста.

`sps_position` (varchar): Посада спеціаліста. Зберігає інформацію про посаду, яку займає спеціаліст.

`sps_phone_number` (varchar): Номер телефону спеціаліста. Зберігає контактний номер телефону спеціаліста.

Таблиця `Specialists` пов'язана з таблицею `Documents` через зовнішній ключ `doc_specialist` в таблиці `Documents`, який посилається на поле `specialist_id` в таблиці `Specialists`. Це забезпечує зв'язок між документами та спеціалістами, дозволяючи системі відстежувати, які спеціалісти пов'язані з кожним

конкретним документом.

Таблиця Material Values є важливою складовою бази даних системи «Розвідка родовищ», оскільки вона зберігає інформацію про матеріальні цінності, які використовуються або видобуваються в процесі розвідки корисних копалин. Ця таблиця дозволяє організувати і зберігати дані про різні матеріали, які можуть бути досліджені або використані в процесі видобутку.

Кожен запис у таблиці має унікальний ідентифікатор, збережений у полі material_id (number, primary key). Це ціле число використовується для унікальної ідентифікації кожного матеріалу в таблиці, що гарантує цілісність даних і дозволяє іншим таблицям посилається на конкретні матеріали через зовнішні ключі.

Поле material_name (varchar) зберігає назву матеріалу. Це текстове поле змінної довжини, яке дозволяє зберігати імена матеріалів будь-якої довжини. Ця інформація є важливою для ідентифікації матеріалу і може використовуватися в різних звітах та аналізах.

Поле material_segment (varchar) містить інформацію про сегмент, до якого належить матеріал. Це також текстове поле змінної довжини, яке дозволяє класифікувати матеріали за різними категоріями або групами. Така класифікація допомагає організувати дані про матеріали і полегшує їх пошук та аналіз.

Поле material_element є зовнішнім ключем, що посилається на таблицю Chemical Elements. Це ціле число вказує на хімічний елемент, з якого складається матеріал або який є важливою складовою частиною матеріалу. Зв'язок із таблицею Chemical Elements дозволяє зберігати додаткову інформацію про хімічні властивості матеріалів, що є важливим для детального аналізу та прийняття науково обґрунтованих рішень у процесі розвідки.

Поле doc_item в таблиці Documents є зовнішнім ключем, що посилається на поле material_id в таблиці Material Values. Цей зв'язок забезпечує зв'язок між документами та матеріальними цінностями, дозволяючи відстежувати, які матеріали використовуються або досліджуються в рамках кожного документа. Це забезпечує більш точне і ефективне управління даними та дозволяє

відстежувати використання матеріалів у процесі розвідки.

Таблиця Chemical Elements призначена для зберігання інформації про хімічні елементи, які є складовими частинами матеріальних цінностей, що використовуються або видобуваються в процесі розвідки корисних копалин. Ця таблиця дозволяє організувати і зберігати дані про різні хімічні елементи, які можуть бути досліджені або використані в процесі видобутку.

Кожен запис у таблиці має унікальний ідентифікатор, збережений у полі `element_id` (number, primary key). Це ціле число використовується для унікальної ідентифікації кожного хімічного елемента в таблиці, що гарантує цілісність даних і дозволяє іншим таблицям посилатися на конкретні елементи через зовнішні ключі.

Поле `element_name` (varchar) зберігає назву хімічного елемента. Це текстове поле змінної довжини, яке дозволяє зберігати назви елементів будь-якої довжини.

Поле `element_type` (varchar) містить інформацію про тип хімічного елемента. Це також текстове поле змінної довжини, яке дозволяє класифікувати елементи за різними категоріями або групами. Така класифікація допомагає організувати дані про елементи і полегшує їх пошук та аналіз.

Поле `element_value` (varchar) зберігає значення або додаткову інформацію про хімічний елемент. Це текстове поле змінної довжини, яке дозволяє зберігати будь-які додаткові дані, пов'язані з елементом. Ця інформація може бути корисною для детального аналізу хімічних властивостей елементів.

Поле `material_element` у таблиці Material Values є зовнішнім ключем, який посилається на поле `element_id` у таблиці Chemical Elements. Це дозволяє забезпечити інтеграцію даних про матеріальні цінності з даними про їх хімічний склад, що дає можливість отримувати детальні дані про хімічні елементи, що входять до складу матеріалів, і проводити їх детальний аналіз. Такий зв'язок є важливим для точного оцінювання якості матеріалів і прийняття обґрунтованих рішень у процесі розвідки.

3.4 Розробка алгоритму функціонування інформаційної системи

Алгоритми – це чітко визначені послідовності інструкцій, призначених для виконання певної задачі або вирішення конкретної проблеми. Вони є основою будь-якої комп'ютерної програми чи системи, оскільки визначають порядок дій, які необхідно виконати для досягнення бажаного результату. Алгоритми забезпечують систематичний підхід до розв'язання задач, що включає визначення вхідних даних, виконання операцій та отримання результатів. Завдяки алгоритмам можливо стандартизувати процеси, підвищити ефективність роботи системи та знизити ймовірність помилок.

Створення моделі алгоритму функціонування інформаційної системи для розвідки родовищ корисних копалин є критичним етапом розробки. Це забезпечує чітке визначення послідовності дій, взаємодії компонентів та ефективного управління інформаційними потоками.

Ретельно спроектований алгоритм дозволяє системі працювати злагоджено та безперебійно, забезпечуючи необхідну функціональність для користувачів.

Одним із визначальних елементів автоматизованих інформаційних систем є блок-схема функціонування, яка відображає структуру системи та взаємозв'язки між її компонентами. Блок-схема є графічним поданням алгоритмів і процедур, що використовуються для обробки даних. Вона допомагає зрозуміти логіку роботи системи, а також визначити можливі місця для оптимізації та вдосконалення. За допомогою блок-схем можна візуально представити процеси, що відбуваються в системі, виявити критичні точки, де можливі збої або неефективність, та розробити шляхи їх усунення.

У підрозділі 3.1 було проведено аналіз вимог до системи та визначено основні функції, які повинна виконувати інформаційна система, включаючи збір, обробку, аналіз даних та управління документами. Відповідно до цього створено блок-схему функціонування інформаційної системи (рисунок 3.8). Потреби користувачів і умови експлуатації системи враховуються для створення гнучкого та адаптивного алгоритму. Алгоритм повинен бути достатньо гнучким, щоб

Надалі користувач за потреби має можливість додати новий запис, редагувати або переглянути існуючий та видалити. Коли користувач бажає додати новий запис, відкривається форма для введення нових даних. Після заповнення форми новий запис зберігається у базі даних і відображається у графічному інтерфейсі.

Якщо користувач бажає редагувати існуючий запис, він обирає його зі списку, після чого відкривається форма з поточними даними запису. Після внесення необхідних змін оновлений запис зберігається у базі даних і відображається на графічному інтерфейсі.

Для перегляду існуючого запису користувач обирає його зі списку, і дані відображаються у детальному вигляді. Видалення запису відбувається після вибору запису та підтвердження видалення. Запис видаляється з бази даних і зникає з графічного інтерфейсу.

При умові, що документи вже існують, користувач може перейти до їх детального перегляду, де дані відображаються у розширеному вигляді, що дозволяє аналізувати й працювати з ними більш ефективно.

На відкритій сторінці документу по полігонах можна побачити відповідний опис матеріальних ресурсів на обраній ділянці. Користувач також має можливість завантажити протокол вимірювання.

Користувач також може переглядати протокол вимірювання безпосередньо у програмі, отримуючи доступ до всіх зафіксованих даних. Редагування протоколу дозволяє оновлювати інформацію про вимірювання, вносити корективи та доповнення, що забезпечує актуальність і точність даних. Видалення протоколу є можливим у випадку, якщо дані більше не актуальні або містять помилки, що потребують повного видалення та повторного введення.

Якщо у протоколі вимірювання зафіксовано показники хімічних ресурсів, що відповідають завчасно заданим параметрам пошуку, автоматично надсилається повідомлення у чат відповідальним особам. Це дозволяє оперативно інформувати відповідальних осіб про важливі показники та забезпечує своєчасне реагування на зміни у даних.

Після внесення й збереження всіх необхідних змін користувач може завершити роботу з програмою, вийшовши з неї. Цей процес докладно показано на діаграмі послідовностей, яка відображає всі кроки для завершення сесії. На блок-схемі цей етап також представлено, що дозволяє побачити логічну послідовність дій та взаємозв'язки між різними компонентами системи. Це забезпечує чітке розуміння завершення роботи програми й гарантує, що всі зміни були збережені належним чином перед виходом.

3.5 Висновки до розділу 3

Проектування архітектури інформаційної системи для розвідки родовищ корисних копалин є складним та багатогранним процесом, що потребує ретельного планування та впровадження. Проведений аналіз дозволяє зробити наступні висновки.

Архітектура системи включає фронтенд, бекенд, базу даних та інтеграційні компоненти, кожен з яких виконує важливі функції для забезпечення ефективної роботи всієї системи. Фронтенд забезпечує зручність та інтуїтивність взаємодії користувача з системою, що є критично важливим для прийняття обґрунтованих рішень на основі зібраних даних. Бекенд відповідає за обробку бізнес-логіки і взаємодію з базою даних, що забезпечує надійність й продуктивність системи.

Використання реляційної моделі бази даних на основі Oracle Database дозволяє ефективно зберігати та обробляти великі обсяги даних, що є важливим для системи розвідки родовищ корисних копалин. Проектування структури бази даних з урахуванням п'яти основних логічних сутностей (документація, типи документів, матеріальні цінності, хімічні елементи, локації, спеціалісти) забезпечує чітку організацію та доступ до даних.

Визначення і врахування функціональних вимог користувачів на ранніх етапах проектування дозволяє створити систему, яка відповідає реальним потребам та очікуванням користувачів. Забезпечення зручного та інтуїтивно зрозумілого інтерфейсу для введення даних сприяє ефективності роботи системи

та задоволеності користувачів.

Автоматизація обробки даних знижує ризик помилок користувача і підвищує загальну надійність й точність системи. Інформаційна система забезпечує різноманітні засоби виведення даних, включаючи автоматичне генерування звітів, візуалізацію даних у вигляді діаграм та графіків, експорт даних.

Блок-схема функціонування системи, що включає всі основні етапи та взаємозв'язки між компонентами, забезпечує чітке розуміння та оптимізацію роботи системи. Наявність алгоритмів для збору, обробки, аналізу даних й управління документами дозволяє ефективно управляти інформаційними потоками та забезпечувати високу продуктивність системи.

Отже, проєктування архітектури інформаційної системи для розвідки родовищ корисних копалин, її технологічних модулів і компонентів є ключовим етапом для створення надійного та ефективного програмного забезпечення, яке здатне виконувати поставлені завдання з високою продуктивністю, надійністю й масштабованістю.

4 РЕАЛІЗАЦІЯ, ОПИС ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ РОЗВІДКИ РОДОВИЩ

У попередніх розділах були розглянуті концептуальні основи проєктування архітектури інформаційної системи для розвідки родовищ корисних копалин, визначені основні вимоги користувачів та розроблені алгоритми її функціонування. Наступним кроком є реалізація цих концепцій у вигляді повноцінної програмної системи, здатної ефективно вирішувати поставлені задачі.

Реалізація інформаційної системи включає кілька важливих етапів: розробку фронтенд і бекенд частин, створення та налаштування бази даних, а також інтеграцію різних компонентів для забезпечення їхньої злагодженої роботи. На цьому етапі важливо враховувати всі аспекти, пов'язані з продуктивністю, безпекою та масштабованістю системи. Кожен з компонентів системи має бути ретельно спроектований і впроваджений відповідно до вимог, визначених на попередніх етапах.

Одним з важливих аспектів реалізації є забезпечення високої якості програмного забезпечення через тестування, що дозволить виявити та виправити помилки, оцінити продуктивність системи та її відповідність функціональним і нефункціональним вимогам.

4.1 Реалізація взаємодії клієнтської та серверної частини

У сучасних веб-додатках інтеграція серверної частини з інтерфейсом користувача є основним чинником, який визначає загальну продуктивність і зручність користування системою. Така взаємодія забезпечує динамічний обмін даними між клієнтом і сервером, дозволяючи оперативно обробляти запити користувачів й надавати їм актуальну інформацію у зручному форматі.

Інтегровані інструменти для розробки веб-додатків, які працюють безпосередньо всередині бази даних, значно спрощують цей процес. Вони

дозволяють розробникам створювати складні програми, що об'єднують у собі інтерфейс користувача, серверну логіку і базу даних у єдиному середовищі. Це дозволяє знизити витрати на розробку, покращити менеджмент проєкту та забезпечити високу продуктивність додатка.

Для взаємодії між серверною частиною та інтерфейсом користувача в системі використовуватимемо Oracle Application Express (APEX). APEX забезпечує зручний та ефективний спосіб розробки веб-додатків на основі бази даних Oracle. Основним розділом APEX є App Builder, який відіграє ключову роль у розробці програм [22]. Тут доступні різноманітні опції для створення та налаштування додатків (рисунок 4.1):

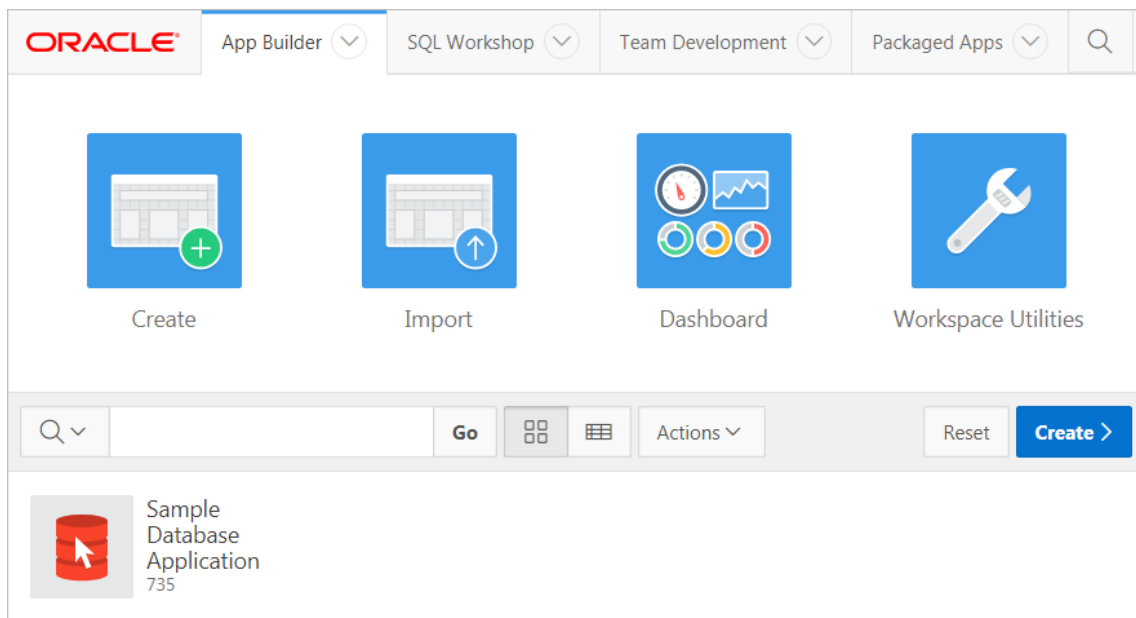


Рисунок 4.1 – Розділ «App Builder» в Oracle Application Express

Створення сторінок: розробники можуть створювати інтерактивні сторінки з використанням вбудованих шаблонів та елементів управління. Це дозволяє швидко створювати зручні та функціональні інтерфейси користувача.

Налаштування макетів: App Builder надає інструменти для налаштування макетів сторінок, включаючи додавання, видалення та редагування регіонів та компонентів.

– Робота з формами: APEX підтримує створення та налаштування

форм для введення та редагування даних. Форми можуть бути пов'язані з таблицями бази даних для автоматичного збереження даних.

- Інтерактивні звіти: розробники можуть створювати інтерактивні звіти для візуалізації та аналізу даних. APEX дозволяє налаштовувати звіти, додаючи фільтри, сортування та групування даних.

- Інтеграція з PL/SQL: APEX підтримує інтеграцію з PL/SQL, що дозволяє використовувати складні логічні обробки та алгоритми безпосередньо у додатках.

- Безпека: APEX надає засоби для налаштування безпеки додатків, включаючи управління доступом користувачів, автентифікацію та авторизацію.

- Динамічні дії: використання динамічних дій дозволяє створювати інтерактивні додатки з реакцією на події, такі як натискання кнопок, зміна значень полів або завантаження сторінок.

На прикладі модуля «Полігони» на рисунку 4.2 показано, як виглядає розділ Page Designer у Oracle Application Express (APEX). У цьому розділі відображається процес налаштування сторінки додатку, де розробники можуть створювати та редагувати елементи інтерфейсу, такі як форми, звіти, кнопки та інші компоненти.

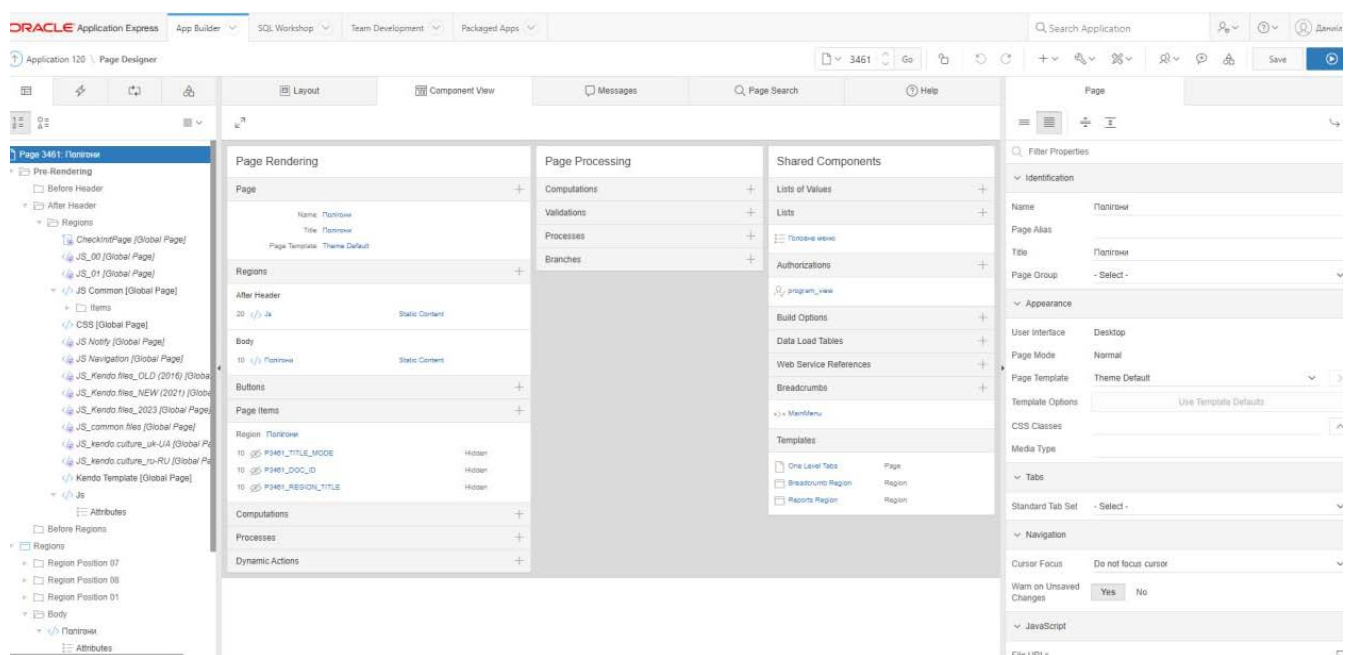


Рисунок 4.2 – Інтерфейс розділу Page Designer

Page Designer забезпечує інтерактивний інтерфейс для побудови веб-сторінок. Він дозволяє швидко вносити зміни та відразу бачити результати. На рисунку показано, як розробники можуть налаштовувати властивості елементів, додавати нові компоненти та налаштовувати їх взаємодію з базою даних через виклики PL/SQL процедур.

Цей приклад ілюструє потужність і гнучкість Oracle APEX у створенні інтерактивних веб-додатків, де інтеграція серверної логіки й користувацького інтерфейсу здійснюється легко та ефективно.

За допомогою APEX розробники можуть викликати PL/SQL процедури для виконання різних завдань, таких як обробка форм, валідація даних, обчислення та інше. Це досягається через використання процесів APEX, які можна налаштувати для виконання певних дій на різних етапах роботи додатка.

APEX підтримує використання технології AJAX (Asynchronous JavaScript and XML), що дозволяє створювати динамічні та інтерактивні інтерфейси. AJAX-запити можуть використовуватися для отримання та відправлення даних на сервер без необхідності перезавантаження сторінки. Це підвищує зручність використання додатка та покращує загальну продуктивність.

На рисунку 4.3 показано взаємодію між клієнтом, Oracle APEX та базою даних Oracle. Клієнт здійснює запити до системи через веб-браузер за протоколом HTTPS. Ці запити передаються до серверної частини, де Oracle APEX обробляє їх за допомогою HTML, CSS та JavaScript для створення інтерактивного інтерфейсу користувача.

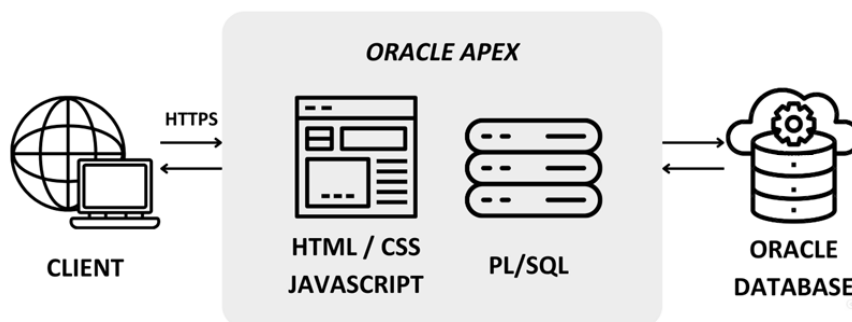


Рисунок 4.3 – Архітектура взаємодії між клієнтом, Oracle APEX та базою даних Oracle

Oracle APEX виконує серверні операції, використовуючи PL/SQL, для взаємодії з базою даних Oracle. Всі запити до бази даних, такі як збереження, оновлення або отримання даних, обробляються через PL/SQL процедури, що забезпечує безпечний та ефективний доступ до даних.

Після обробки запитів результати повертаються клієнту, вбезпечуючи зручний та інтерактивний користувацький досвід. Рисунок 4.3 ілюструє, як Oracle APEX забезпечує тісну інтеграцію між клієнтською частиною, серверною логікою та базою даних, створюючи потужні веб-додатки.

Незалежно від того, чи користувач створює новий застосунок, чи працює в існуючому середовищі розробки, процеси в базі даних залишаються незмінними. Браузер ініціює запит за допомогою URL, який обробляється системою APEX з використанням PL/SQL. База даних відповідає на цей запит та відправляє результат у вигляді HTML сторінки назад у браузер. Ці дії виконуються щоразу при надходженні запиту або завантаженні сторінки.

Система APEX керує станом сесії програми безпосередньо в базі даних, не потребуючи спеціального постійного з'єднання. Кожен запит обробляється в окремій сесії бази даних, що дозволяє оптимально використовувати ресурси процесора.

Завдяки цим можливостям Oracle Application Express (APEX) забезпечує ефективну взаємодію між серверною частиною та інтерфейсом користувача, спрощуючи процес розробки та підтримки веб-додатків на базі Oracle.

4.2 Реалізація алгоритмів обробки даних у базі даних

Реалізація алгоритмів обробки даних у базі даних є ключовим етапом у розробці інформаційної системи для розвідки родовищ корисних копалин. Цей процес включає розробку та впровадження різних алгоритмів для забезпечення ефективного збору, зберігання, аналізу та обробки даних. У цьому підрозділі розглянемо основні аспекти реалізації таких алгоритмів за допомогою мови програмування PL/SQL.

Як попередньо зазначено, для реалізації алгоритмів обробки даних було обрано платформу Oracle Database, яка забезпечує високу продуктивність та надійність. Основними інструментами для розробки алгоритмів стали PL/SQL – процедурне розширення для SQL, яке дозволяє створювати складні програми всередині бази даних.

Реалізація алгоритмів збору даних включала розробку PL/SQL процедур та тригерів для імпорту даних з різних джерел, таких як геологічні звіти, результати польових досліджень та дані з геофізичних приладів. Ці процедури забезпечують коректний імпорт даних у базу даних, виконуючи перевірку та валідацію отриманої інформації.

Алгоритми обробки даних реалізовані за допомогою PL/SQL пакетів, які включають складні запити та логіку для виконання операцій фільтрації, агрегації та аналізу. Для підвищення продуктивності використовувалися індекси та оптимізація запитів.

У системі передбачено можливість додавання нових записів до бази даних. Ця функція реалізована за допомогою спеціальних форм, де користувачі мають можливість вводити необхідну інформацію. Після заповнення форми дані передаються на сервер, де PL/SQL процедури здійснюють їх перевірку та вставку у відповідні таблиці бази даних. Додавання нових даних забезпечує постійне оновлення і розширення інформаційної бази системи.

Система також дозволяє користувачам редагувати існуючі записи. Для цього розроблені спеціальні інтерфейси, які відображають поточні дані та дозволяють вносити зміни. Після внесення змін вони передаються на сервер, де PL/SQL процедури здійснюють перевірку та оновлення відповідних записів у базі даних. Це забезпечує актуальність і точність інформації в системі.

Система підтримує регулярне оновлення даних для забезпечення їх актуальності. Оновлення може включати імпорт нових даних, ручне або автоматичне оновлення на основі заданих правил. Процедури PL/SQL здійснюють обробку та оновлення даних у базі, забезпечуючи високу продуктивність та надійність системи.

Для видалення застарілих або непотрібних записів у системі реалізовані механізми безпечного видалення даних. Користувачі можуть вибирати записи для видалення через інтерфейс, після чого система запитує підтвердження дії. Після підтвердження, PL/SQL процедури видаляють обрані записи з бази даних. Це дозволяє підтримувати базу даних в актуальному стані та уникати накопичення зайвої інформації.

Всі ці операції – додавання, редагування, видалення та оновлення даних - інтегровані у систему для забезпечення повного циклу обробки даних. Вони гарантують, що інформація в базі даних завжди актуальна, точна та готова до використання у будь-який момент.

На рисунку 4.4 зображено цикл цих процесів, які виконуються послідовно для забезпечення ефективного управління розвідкою родовищ корисних копалин.



Рисунок 4.4 – Цикл процесів розвідки у інформаційній системі

Процес додавання даних починається зі створення документа, який визначає конкретне місце розвідки, позначене як квадрат. До цього документа

додається загальна інформація про квадрат, включаючи геологічні дані та інші релевантні відомості. Це забезпечує повний і точний запис у базі даних.

Система дозволяє редагувати існуючі записи, що забезпечує актуалізацію даних. Редагування може включати оновлення інформації про місце розвідки, зміну деталей протоколу розвідки та інші корективи.

Для видалення застарілих або непотрібних записів передбачені механізми безпечного видалення. Це дозволяє підтримувати базу даних в актуальному стані та уникати накопичення зайвої інформації.

Процес оновлення даних передбачає регулярне оновлення статусу розвідки, аналіз протоколу та інформування відповідальних осіб про результати розвідки. Це забезпечує постійне оновлення бази даних та підтримання її в актуальному стані.

4.3 Реалізація користувацького інтерфейсу

Для реалізації користувацького інтерфейсу було обрано сучасні веб-технології, які забезпечують високу продуктивність, гнучкість та зручність у використанні. Основними технологіями стали HTML для створення структури сторінок, CSS для стилізації елементів інтерфейсу та JavaScript для додавання інтерактивності. Використання бібліотеки Kendo UI дозволило значно прискорити розробку завдяки набору готових компонентів, таких як таблиці, графіки, форми тощо.

Процес реалізації основних компонентів користувацького інтерфейсу розпочато зі створення кількох ключових сторінок, включаючи головну сторінку, сторінку введення та обробки даних, а також сторінку звітності. Кожен компонент розроблявся з урахуванням можливості повторного використання, що дозволяє спростити подальшу підтримку та розширення системи.

Форми для введення даних були реалізовані з використанням HTML та CSS, що забезпечує швидкий рендеринг і відповідність вимогам до стилізації. Для валідації введених даних у режимі реального часу було використано

JavaScript, що дозволяє запобігти введенню некоректної інформації.

Головний інтерфейс інформаційної системи для розвідки родовищ корисних копалин призначений для зручного доступу до основних функцій системи, а також для ефективного управління даними.

Інтерфейс складається з кількох основних областей: заголовков, пошукова область із фільтром, область кнопок управління та таблична область для відображення даних (рисунк 4.5).

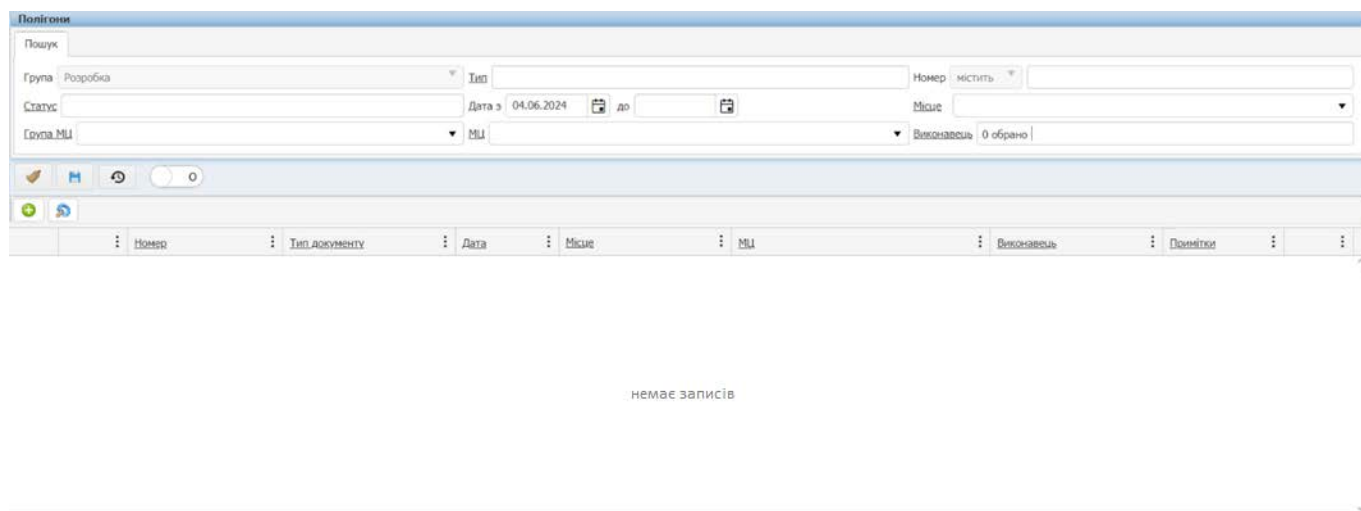


Рисунок 4.5 – Головний інтерфейс додатку

Після натискання кнопки «Пошук» запускається процес, який здійснює фільтрацію даних за всіма введеними параметрами. Після обробки введених критеріїв система виводить відповідну інформацію на головний інтерфейс, забезпечуючи користувачу доступ до результатів пошуку (рисунк 4.6).

Номер	Тип документа	Дата	Місце зберігання	МЦ	Виконавець	Примітки
РЗХВ-2312-006	Буріння	29.12.23	сектор II A	концентрат германію	Коваленко І. І.	Тест
РЗХВ-2312-001	Буріння	28.12.23	квадрат 1:1	ванадієвий кек	Юрчук М. С.	тестування
РЗХВ-2312-004	Копання ручне	29.12.23	квадрат 1:1	ванадієвий кек	Тестувальник К. К.	
РЗХВ-2312-005	Копання ручне	29.12.23	квадрат 1:1	ванадієвий кек	Іваненко О. О.	1
РЗХВ-2312-002	Копання ручне	29.12.23	квадрат 1:1	плав хлоратора	Юрчук М. С.	тест

Рисунок 4.6 – Результати пошуку по фільтрах

Пошукова область розташована під заголовком і призначена для введення критеріїв пошуку даних. Вона складається з кількох полів для введення та вибору, що дозволяють користувачам швидко знаходити необхідну інформацію в системі.

Пошукова область включає випадаючий список для вибору групи даних, що дозволяє фільтрувати записи за певними категоріями. Також є випадаючий список для вибору типу документів, що допомагає звужити результати пошуку до певного типу розвідки (рисунок 4.7):

- буріння;
- копання механізоване;
- копання вручну.

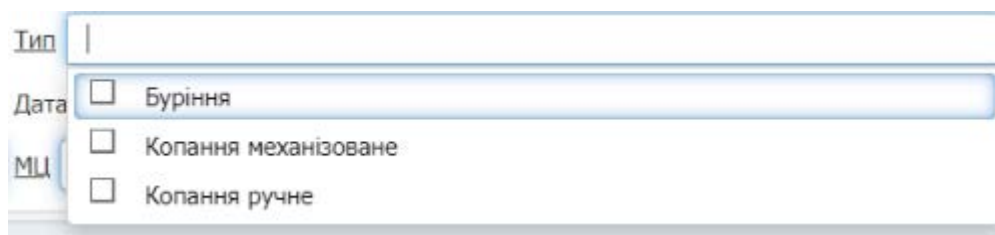


Рисунок 4.7 – Випадаючий список компоненту «Тип»

Для більш точного пошуку є поле для введення номеру документа, котре дозволяє відшукати конкретний запис за його унікальним номером. Пошук можливий бути за повною відповідністю номеру або ж за частковим його вмістом, наприклад, за ключовими словами або фразами.

Поле «Статус» зі списком дозволяє користувачам обирати статус документа з попередньо визначених варіантів (рисунок 4.8): анульовано, виконано, виконується і редагується. Статус «анульовано» вказує на те, що документ було скасовано і більше не є актуальним. Статус «виконано» означає, що розвідку було завершено або документ оброблено. Статус «виконується» позначає, що розвідка або документ знаходиться в процесі виконання. Статус «редагується» вказує на те, що документ або завдання знаходиться на етапі редагування в інформаційній системі.

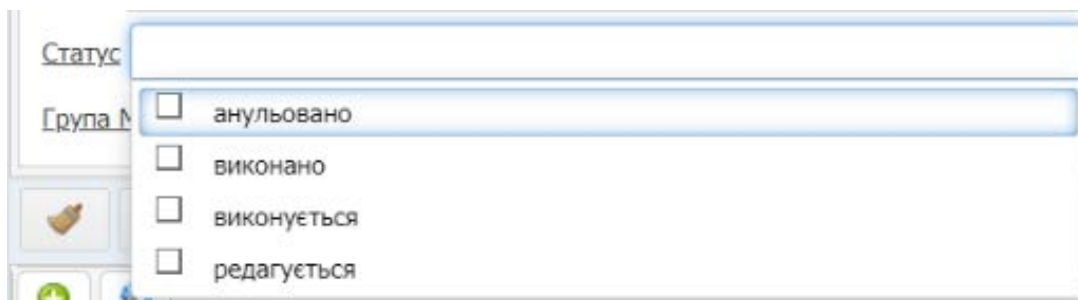


Рисунок 4.8 – Випадаючий список компоненту «Статус»

Також пошуковий фільтр містить поле для вибору діапазону дат, що дозволяє користувачам фільтрувати дані в системі за певним часовим проміжком. Поле складається з двох частин: поле «Дата з» і поле «до».

Поле «Дата з» дозволяє користувачеві вибрати початкову дату діапазону. Користувач може або вручну ввести дату у форматі дд.мм.рррр, або скористатися вбудованим календарем для зручного вибору дати. Натиснувши на іконку календаря праворуч від поля, користувач відкриває календар, де можна вибрати потрібну дату.

Поле «до» дозволяє вказати кінцеву дату діапазону. Як і у випадку з полем «Дата з», користувач може ввести дату вручну або скористатися вбудованим календарем. Іконка календаря поруч із полем також відкриває календар для вибору дати.

Використання цих полів для дат разом дозволяє точно визначити часовий проміжок, за який необхідно знайти дані. Це особливо корисно для аналізу інформації, яка має часові обмеження або залежить від певних періодів. Фільтрація за діапазоном дат допомагає користувачам швидко знаходити потрібні записи та аналізувати дані в контексті обраного періоду.

Поле «Місце» є важливим елементом пошукової області інтерфейсу інформаційної системи для розвідки родовищ корисних копалин (рисунок 4.9). Воно призначене для вибору конкретного місця або ділянки, що підлягає дослідженню або обробці.

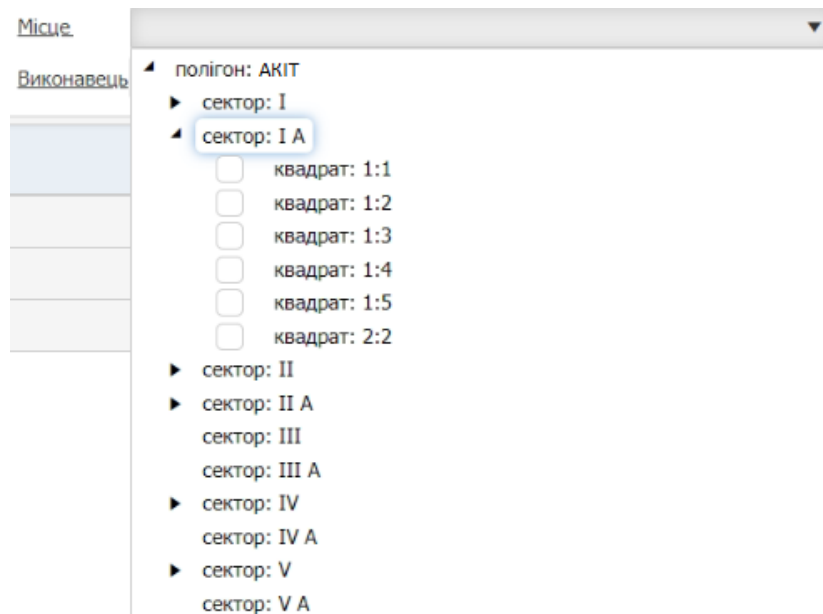


Рисунок 4.9 – Випадаючий список компоненту «Місце»

Поле «Місце» використовує ієрархічну структуру та організоване у вигляді дерева з кількома рівнями вкладеності. На верхньому рівні знаходиться найзагальніша категорія, наприклад, полігон «АКІТ». Нижче розташовані підкатегорії, такі як сектори та квадрати, що дозволяють більш точно визначити місце.

Користувач може натиснути на стрілочку поруч із назвою категорії, щоб розгорнути або згорнути список підкатегорій. Це дозволяє поступово деталізувати вибір, починаючи з більш загальних локацій і переходячи до конкретних місць. Кожен рівень ієрархії представлений у вигляді списку елементів, з яких можна обирати. Це забезпечує зручну навігацію по ієрархічній структурі, дозволяючи користувачам швидко знаходити та обирати необхідні локації. Це особливо корисно для великих полігонів або комплексних географічних об'єктів, де важливо точно визначити місце проведення робіт або зберігання документів та знизити час, витрачений на пошук потрібних локацій.

Поле «Група МЦ» в інформаційній системі призначене для вибору групи матеріально-технічних цінностей (МЦ), до яких належать різні типи матеріалів та відходів (рисунок 4.10). Це поле є важливим елементом пошукової області, що дозволяє користувачам фільтрувати дані за конкретною групою МЦ.

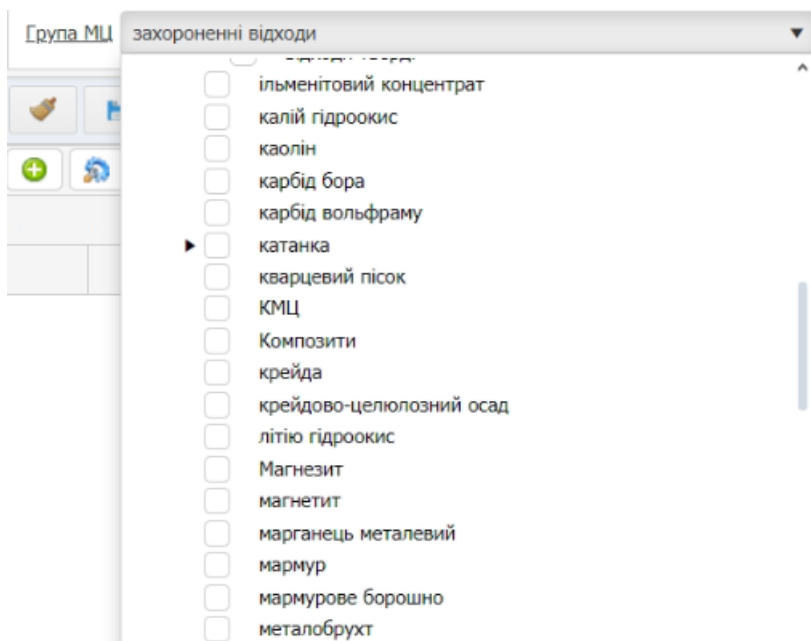


Рисунок 4.10 – Випадаючий список компоненту «Група МЦ»

Поле «Група МЦ» організоване у вигляді випадаючого списку, який містить категорії матеріально-технічних цінностей. Користувач може натиснути на поле, щоб відкрити список, і вибрати необхідну категорію. У даному прикладі обрана група «заховані відходи». Після вибору групи список розгортається, показуючи підкатегорії та елементи, що належать до цієї групи, такі як «катанка», «кварцевий пісок», «каолін» тощо. Це забезпечує зручний спосіб для користувачів швидко знаходити потрібні матеріали чи відходи за їх типом.

Інформаційна система може бути використана не тільки для первинної розвідки, але й для управління відходами з невідомим складом. Поле «Група МЦ» дозволяє класифікувати такі відходи, забезпечуючи можливість їх ідентифікації та подальше дослідження. Це забезпечує систематичний підхід до управління матеріально-технічними цінностями, підвищуючи точність і швидкість пошуку необхідної інформації. Це поле є важливим інструментом для організації та обробки великих обсягів даних, пов'язаних з різними типами матеріалів та відходів.

Поле «МЦ» призначене для детального вибору та пошуку матеріально-технічних цінностей. Воно організоване у вигляді випадаючого списку з можливістю розгортання підкатегорій, що дозволяє користувачам переглядати

та вибирати конкретні елементи. Вибір у цьому полі здійснюється через ієрархічну структуру, що допомагає систематизувати дані та забезпечити легкий доступ до них (рисунок 4.11).

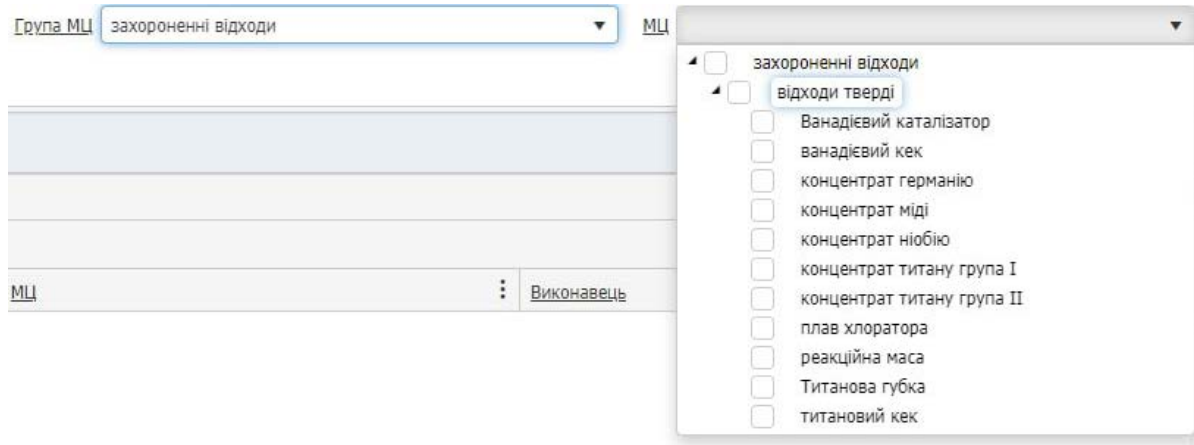


Рисунок 4.11 – Випадаючий список компоненту «МЦ»

Поле «МЦ» є особливо корисним для класифікації відходів з невідомим складом. Воно дозволяє систематично організовувати відходи за різними категоріями та підкатегоріями, що спрощує їх ідентифікацію та подальшу обробку. Наприклад, відходи можуть бути класифіковані за типами, такими як «тверді відходи», і далі деталізовані до конкретних видів відходів.

Поле «Виконавець» в інформаційній системі призначене для вибору або пошуку відповідальної особи, яка виконує певне завдання або працює з конкретним документом. Це поле є важливим елементом пошукової області інтерфейсу, забезпечуючи точний і швидкий доступ до інформації про виконавців. Поле реалізовано у вигляді випадаючого списку з можливістю введення тексту для пошуку. Користувач може почати вводити ім'я виконавця, і система автоматично запропонує відповідні варіанти. Це дозволяє швидко знаходити необхідну особу без необхідності прокручувати весь список.

Наприклад, введення слова «тест» у полі «Виконавець» призводить до відображення списку всіх виконавців, чийі імена або пов'язані дані містять цей текст (рисунок 4.12). У списку можуть бути такі записи, як «Кадровий Тест [Тестую ХК]», «Тест Павло Петрович [КонтрАгенти]» та інші. Це забезпечує

зручний спосіб вибору виконавця для конкретного завдання або документа.

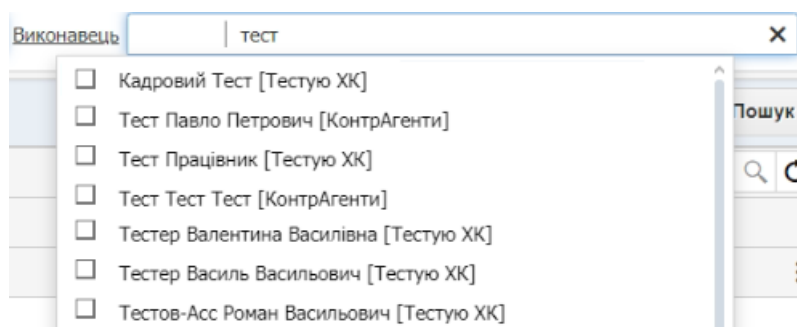


Рисунок 4.12 – Випадаючий список компоненту «Виконавець»

Поле «Виконавець» підтримує фільтрацію та сортування даних, що підвищує ефективність роботи користувачів. Користувач може вибрати одного або кілька виконавців, відзначаючи відповідні записи в списку. Це особливо корисно для завдань, які вимагають участі кількох осіб або потребують розподілу відповідальності.

Для додавання документу передбачено відповідно кнопку під пошуковим фільтром. При її натисканні відкривається модальне вікно із заголовком «Полігони: додавання» (рисунок 4.13).

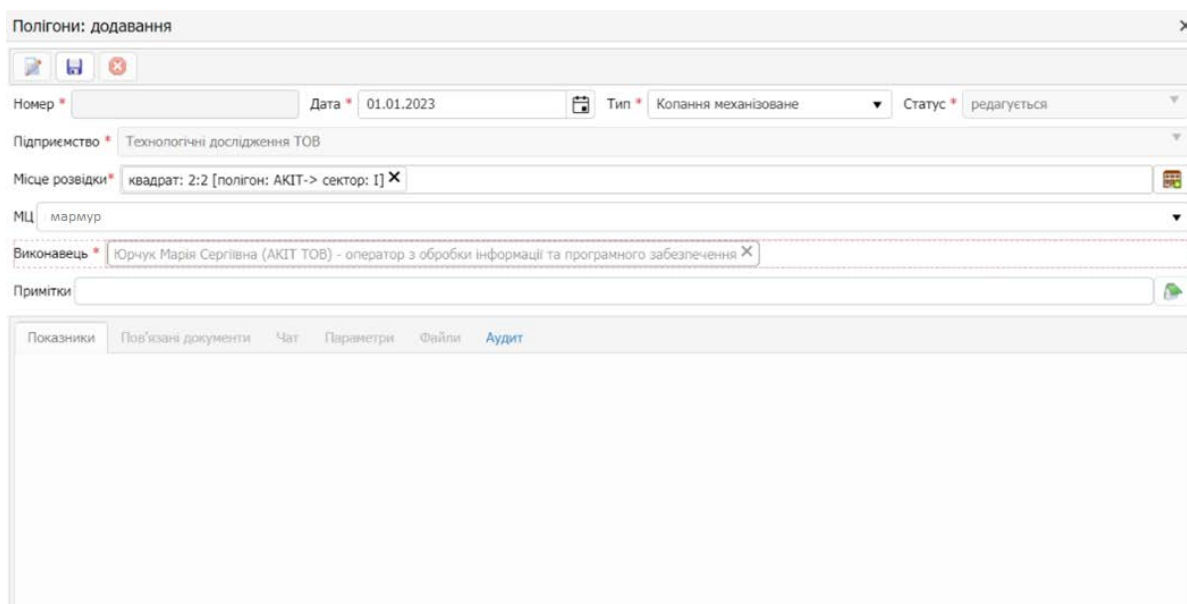


Рисунок 4.13 – Інтерфейс додавання документу по полігону

Інтерфейс «Полігони: додавання» має декілька ключових полів, які забезпечують введення та редагування даних, необхідних для управління процесом розвідки.

Поле «Дата» дозволяє вказати дату початку розвідки та створення документу за допомогою вбудованого календаря. Вибір типу розвідки здійснюється через випадаюче меню «Тип», де для прикладу обрано тип «Копання механізоване».

Поле «Підприємство» призначене для введення або вибору підприємства, яке здійснює розвідку, і в прикладі зазначено «Технологічні дослідження ТОВ». Місце розвідки вказується у полі «Місце розвідки», де в прикладі зазначено «квадрат: 2:2 [полігон: АКІТ -> сектор: 1]». Поле «МЦ» використовується для вибору матеріальних цінностей, які очікуються на місці розвідки, і в даному випадку обрано «мармур».

Якщо ж потрібного місця розвідки у базі даних не існує, то його можна створити. Для цього користувачу доступна кнопка зправа від поля «Місце розвідки». При її натисканні відкривається додаткове вікно, інтерфейс якого можна побачити на рисунку 4.14.

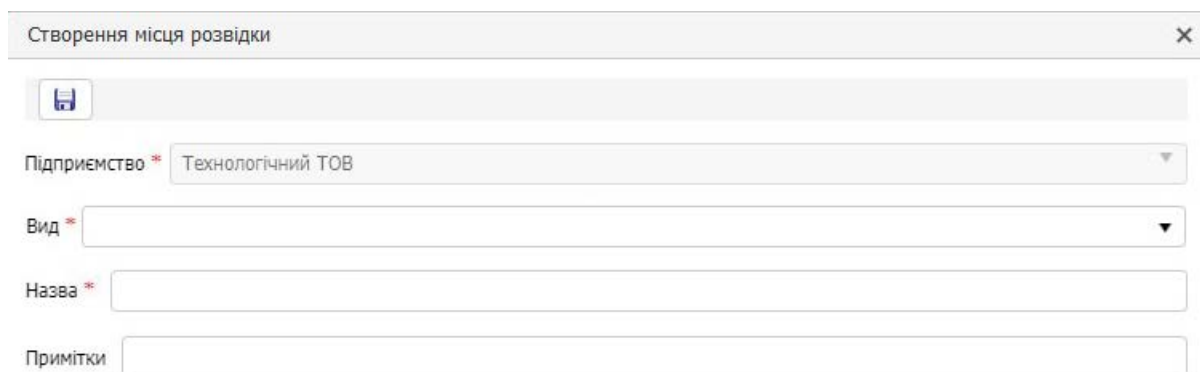


Рисунок 4.14 – Інтерфейс вікна створення місця розвідки

Поле «Виконавець» у інтерфейсі створення документу дозволяє обирати відповідального виконавця, вказуючи його ім'я та посаду. У прикладі зазначено «Юрчук Марія Сергіївна (АКІТ ТОВ) - оператор з обробки інформації та програмного забезпечення». Поле «Примітки» надає можливість додати

додаткову інформацію або коментарі, пов'язані з розвідкою.

Внизу вікна розміщені вкладки, котрі стануть доступними лише після створення документу.

Частина полів позначені зірочками. Ці зірочки вказують на те, що поля є обов'язковими для заповнення. Обов'язкові поля забезпечують введення мінімально необхідної інформації для створення коректного та повного запису про розвідку полігону.

Наприклад, поле «Підприємство» позначене зірочкою, що означає необхідність вказати назву підприємства, яке здійснює розвідку. Без цієї інформації запис не може бути створений, оскільки не буде відомо, яка організація відповідає за проведення розвідувальних робіт.

Поле «Місце розвідки» також позначене зірочкою. Це важливе поле, оскільки вказує точне місце, де проводиться розвідка. Введення цієї інформації є критично важливим для ідентифікації та управління різними місцями розвідки.

Поле «Виконавець» обов'язкове для заповнення, оскільки вказує на відповідальну особу, яка виконує розвідувальні роботи. Це поле дозволяє призначити відповідального за проведення робіт і забезпечує контроль за виконанням завдань.

Загалом, позначення полів зірочками допомагає користувачам швидко ідентифікувати обов'язкові для заповнення поля, що сприяє правильному введенню даних та запобігає виникненню помилок під час створення нового запису в системі.

Інтерфейс створеного документу містить декілька важливих елементів, які забезпечують управління даними та виконання основних операцій. Цей інтерфейс забезпечує зручний спосіб введення, зберігання та управління даними про механізоване копання, надаючи користувачам можливість швидко створювати та редагувати документи, а також виконувати інші необхідні операції. Його приклад зображено на рисунку 4.15, де заголовок вікна вказує тип документа – «Копання механізоване», його унікальний згенерований системою номер «РЗзхВ-2301-001» та дату створення «01.01.2023»; статус документа

позначений як «редагується», що означає, що даний запис ще знаходиться в процесі редагування і не був остаточно збережений.

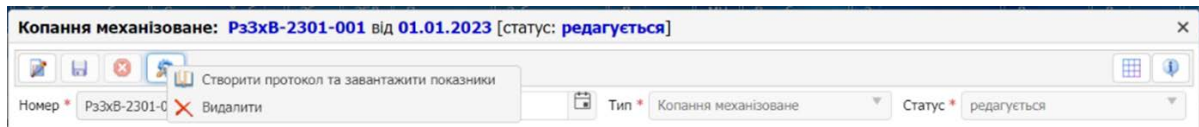


Рисунок 4.15 – Частина інтерфейсу перегляду та редагування документу розвідки

У верхній частині інтерфейсу розташовані кнопки для основних операцій:

- Кнопка збереження (піктограма дискети) дозволяє зберегти зміни в документі.
- Кнопка видалення (піктограма червоного хреста) призначена для видалення документа.
- Кнопка створення протоколу та завантаження показників (піктограма документа з зеленим плюсом) відкриває меню з опціями «Створити протокол та завантажити показники» і «Видалити». Ці опції дозволяють користувачеві створювати новий протокол розвідки або видаляти існуючий запис.

Інтерфейс «Прикріплення файлів» відкривається при натисканні на кнопку «Створити протокол та завантажити показники» (рисунок 4.16). Цей інтерфейс дозволяє користувачам завантажувати файли, які містять додаткову інформацію або показники, пов'язані з конкретним документом про розвідку.

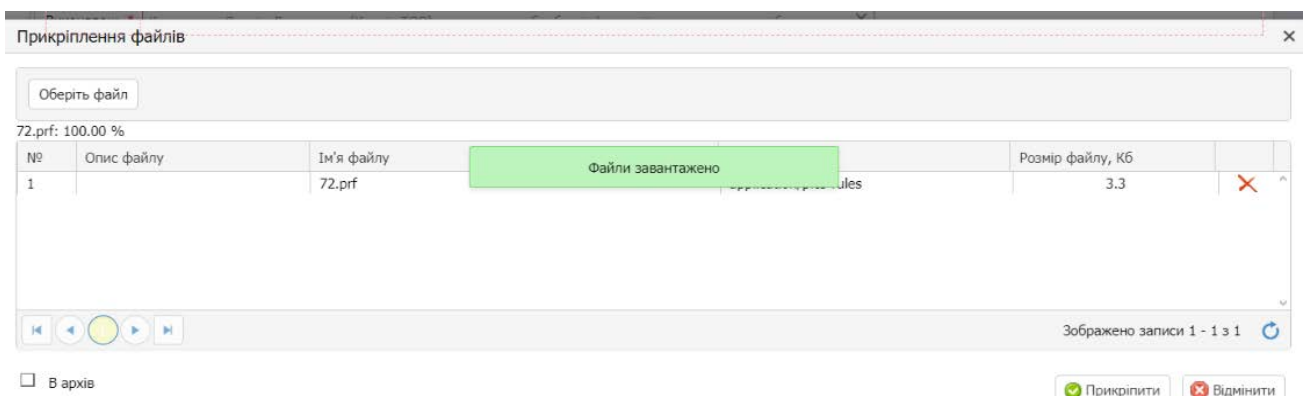


Рисунок 4.16 –Інтерфейс прикріплення файлу з протоколом розвідки

Верхня частина вікна «Прикріплення файлів» містить кнопку «Обріть файл», за допомогою якої користувач може вибрати файл для завантаження з локального комп'ютера. Після вибору файлу відображається його ім'я та індикатор прогресу завантаження, який показує відсоток завантаженого файлу.

Нижче розташована таблиця, яка відображає інформацію про завантажені файли. В таблиці представлені наступні стовпці: № (порядковий номер файлу у списку), опис файлу, ім'я файлу, формат файлу (визначений MIME-типом), розмір файлу.

Кожен рядок таблиці містить інформацію про окремий файл. У прикладі показано файл з ім'ям «72.prf», форматом «application/pics-rules» та розміром 3.3 КБ. Справа від кожного рядка є кнопка з червоним хрестиком, яка дозволяє видалити файл зі списку завантажених.

У нижній частині вікна розташовані кнопки навігації для перемикання між сторінками файлів, якщо їх більше ніж одна сторінка. Також є чекбокс «В архів», який дозволяє вказати, чи слід переміщувати файл до архіву.

Внизу інтерфейсу прикріплення протоколу розвідки розміщені дві основні кнопки:

- Прикріпити: Кнопка для збереження завантажених файлів та прикріплення їх до документа.
- Відмінити: Кнопка для скасування дії та закриття вікна без збереження змін.

Цей інтерфейс забезпечує зручний спосіб завантаження та управління файлами, пов'язаними з протоколами розвідки, що допомагає зберігати та організовувати додаткову інформацію для ефективного управління процесом розвідки.

При натисканні на кнопку «Прикріпити» ініціалізується процес обробки файлу. Після завершення обробки файлу користувач може побачити вікно з інформацією щодо хімічних елементів, які не було знайдено у протоколі (рисунок 4.17). Інформація відображено у вигляді таблиці з наступними стовпцями:

– Текст протоколу – цей стовпець містить повідомлення про хімічні елементи, які не були знайдені. Наприклад, «Не знайдено AlO в протоколі», «Не знайдено Ba в протоколі» і т.д.

– Дата – у цьому стовпці вказана дата та час, коли був створений запис про неznайдені елементи.

Текст протоколу	Дата
Не знайдено AlO в протоколі	11.06.2024 15:17:08
Не знайдено Ba в протоколі	11.06.2024 15:17:08
Не знайдено BaO в протоколі	11.06.2024 15:17:08
Не знайдено CrO в протоколі	11.06.2024 15:17:08
Не знайдено CuO в протоколі	11.06.2024 15:17:08
Не знайдено FeO в протоколі	11.06.2024 15:17:08
Не знайдено NbO в протоколі	11.06.2024 15:17:08
Не знайдено SO в протоколі	11.06.2024 15:17:08
Не знайдено SiO в протоколі	11.06.2024 15:17:08
Не знайдено TiO в протоколі	11.06.2024 15:17:08
Не знайдено VO в протоколі	11.06.2024 15:17:08
Не знайдено ZrO в протоколі	11.06.2024 15:17:08

Рисунок 4.17 –Інтерфейс інформації по незнайдених хімічних елементах у протоколі

У нижній частині вікна розташовані дві основні кнопки:

– Далі – кнопка дозволяє продовжити роботу з протоколом, незважаючи на виявлені відсутні хімічні елементи.

– Закрити – кнопка закриває вікно, дозволяючи користувачеві переглянути або виправити протокол перед продовженням.

Розглянувши на рисунку 4.17 вікно, забезпечує користувача важливою інформацією про потенційні проблеми з протоколом, дозволяючи йому вчасно внести необхідні корективи або продовжити роботу з урахуванням цих попереджень.

Після завершення обробки файлу з протоколом, система автоматично інтегрує значення з протоколу у основний файл по розвідці. Це дозволяє оперативно оновлювати дані та забезпечувати актуальність інформації.

На прикладі вікна «Копання механізоване» на рисунку 4.18 показано, як значення з протоколу підтягуються до відповідних полів основного файлу.

Нижче основних полів документу наведено табличний розділ, що містить вкладки з додатковими даними.

Копання механізоване: РЗЗХВ-2301-001 від **01.01.2023** [статус: **редагується**]

Номер * РЗЗХВ-2301-001 Дата * 01.01.2023 Тип * Копання механізоване Статус * редагується

Підприємство * Технологічні дослідження ТОВ

Місце розвідки * квадрат: 2:2 [полігон: АКІТ -> сектор: 1] X

МЦ концентрат ніобію

Виконавець * Юрчук Марія Сергіївна (АКІТ ТОВ) - оператор з обробки інформації та програмного забезпечення X

Примітки

Показники Пов'язані документи Чат Параметри Файли Аудит

№	Місце	Сор...	Назва	Одиниці вимірюв...	Тип пара...	Тип пере...	Мінімал...	Максим...	Значення
178		Sm	2O3	відсоток	шаблон	м'яка			
179		Cd	1O1	відсоток	шаблон	м'яка			
Місце: №6 [проба № 6]									
6		C		відсоток	шаблон	м'яка			
8		O		відсоток	шаблон	м'яка			40.311
12		Mg		відсоток	шаблон	м'яка			1.358
13		Al		відсоток	шаблон	м'яка			3.623
14		Si		відсоток	шаблон	м'яка			8.504
15		P		відсоток	шаблон	м'яка			
16		S		відсоток	шаблон	м'яка			0.667
17		Cl		відсоток	шаблон	м'яка			3.033

Рисунок 4.18 –Відображення інформації із протоколу у основному документі

У вкладці «Показники» відображаються значення, які були автоматично підтягнуті з протоколу. Таблиця містить наступні стовпці: порядковий номер, назва показника, одиниці вимірювання, тип параметра, тип передавання, мінімальні та максимальні значення, знайдені значення.

У прикладі на рисунку 4.18 таблиця містить показники для місця №6 [проба №6], де для кожного хімічного елемента (наприклад, C, O, Mg, Al, Si, P, S, Cl) вказані відповідні значення у відсотках, що були автоматично імпортовані з протоколу.

Якщо знайдене значення відповідає шуканому, то така комірка виділяється зеленим кольором (рисунок 4.19). Це дозволяє користувачам швидко ідентифікувати значення, які відповідають встановленим критеріям. Зелене виділення комірки є візуальним індикатором того, що знайдене значення

успішно пройшло перевірку відповідності, що значно спрощує аналіз даних та прийняття рішень. Це також допомагає уникнути помилок, пов'язаних з ручною перевіркою, підвищуючи точність і ефективність роботи з даними у системі розвідки родовищ.

Сор...	Назва	Одиниці вимірюв...	Тип пара...	Тип пере...	Мінімал...	Максим...	Значення
19	K	відсоток	шаблон	м'яка			
20	Ca	відсоток	шаблон	м'яка			
21	Sc	відсоток	сертифікат	жорстка	0	0.01	
22	Ti	відсоток	сертифікат	жорстка	0	5	29.283
23	V	відсоток	сертифікат	жорстка	0	2.5	0.71
24	Cr	відсоток	сертифікат	жорстка	0	5	3.233

Рисунок 4.19 – Приклад виділення шуканого значення у таблиці

Окрім того, це означає, що розвідка досягла цілі і бажаний елемент було знайдено. Про це у чаті інформується попередньо назначена відповідальна особа. У прикладі вікна чату показано, що повідомлення про знайдений елемент відправляється адресатам (рисунок 4.20).

Показники Пов'язані документи Чат Параметри Файли Аудит							
		Текст	Адресати	Користувач	Дата ↑		
		У квадрат: 2:2 [полігон: ЗТМ -> сектор: I] матеріальної цінності концентрат ніобію 006 [біг-бег] знайдено: Ti- 29.283%	Кузьменко А. В.; Полі...	Кузьменко Д. Д.	11.06.24 15:17:33		

Рисунок 4.20 – Приклад відправленого повідомлення

Повідомлення містить деталі про місце розвідки та знайдений елемент, наприклад: «У квадраті: 2:2 [полігон: ЗТМ -> сектор: 1] матеріальної цінності концентрат ніобію 006 [біг-бег] знайдено: Ti - 29.283%». Це повідомлення відображається у системі у вигляді спливаючої нотифікації при вході адресата в систему, що забезпечує оперативне інформування про важливі.

При натисканні на зелену іконку у вкладці «Чат» відкривається вікно «Адресати», котре використовується для управління списком осіб, яким будуть надсилатися повідомлення про результати розвідки. За замовчуванням повідомлення надсилається відповідальній особі, яка була вказана при створенні

документа, та інженеру-технологу, що завантажував протокол (рисунок 4.21).

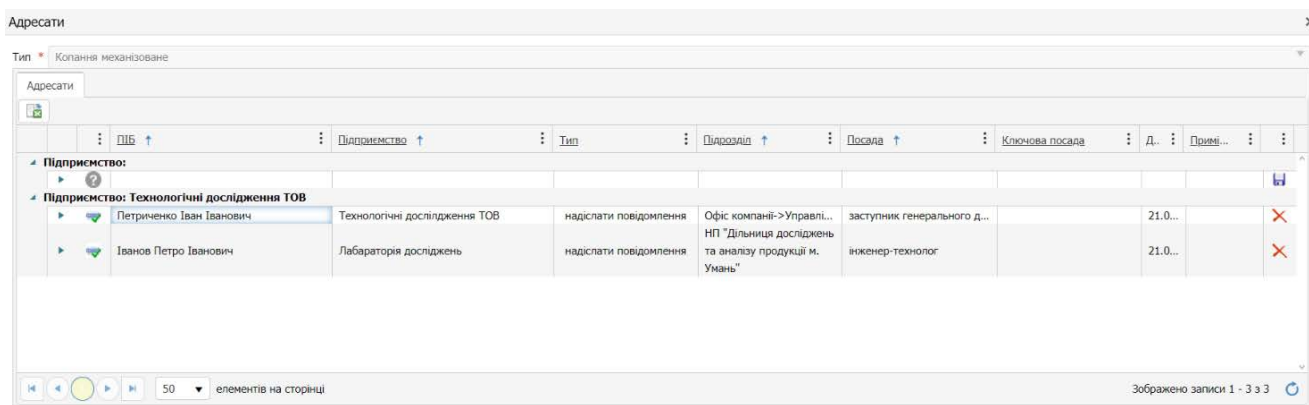


Рисунок 4.21 – Інтерфейс вікна «Адресати»

У нижній частині вікна розташовані кнопки навігації для перемикавання між сторінками адресатів, якщо їх більше ніж одна сторінка. Користувач може вибрати кількість записів, що відображаються на сторінці, за допомогою випадаючого меню.

Це вікно забезпечує гнучкість у налаштуванні списку адресатів для надсилання повідомлень, що дозволяє оперативно інформувати відповідальних осіб про результати розвідки та інші важливі події.

Окрім автоматичної розсилки повідомлень про аналіз, користувачі мають можливість використовувати чат для обміну інформацією по документу. Чат дозволяє оперативно спілкуватися з колегами, обговорювати деталі для вношення зміни у реальному часі.

На рисунку 4.X відображені результати всіх розвідок у квадраті 7:22 у секторі I на полігоні ЗТМ. Вікно складається з двох основних частин: таблиці та графіку.

Таблиця, розташована ліворуч, містить інформацію про концентрації різних хімічних елементів у вибраних пробах. Для кожного елемента вказані мінімальні, максимальні та середні значення, що дозволяє отримати повну картину розподілу елементів у дослідженій області.

Графік, розташований праворуч, візуалізує ці дані, показуючи динаміку

змін концентрацій хімічних елементів у різних пробах. Графічний формат дозволяє швидко ідентифікувати тренди та аномалії, що може бути корисним для подальшого аналізу та планування робіт.

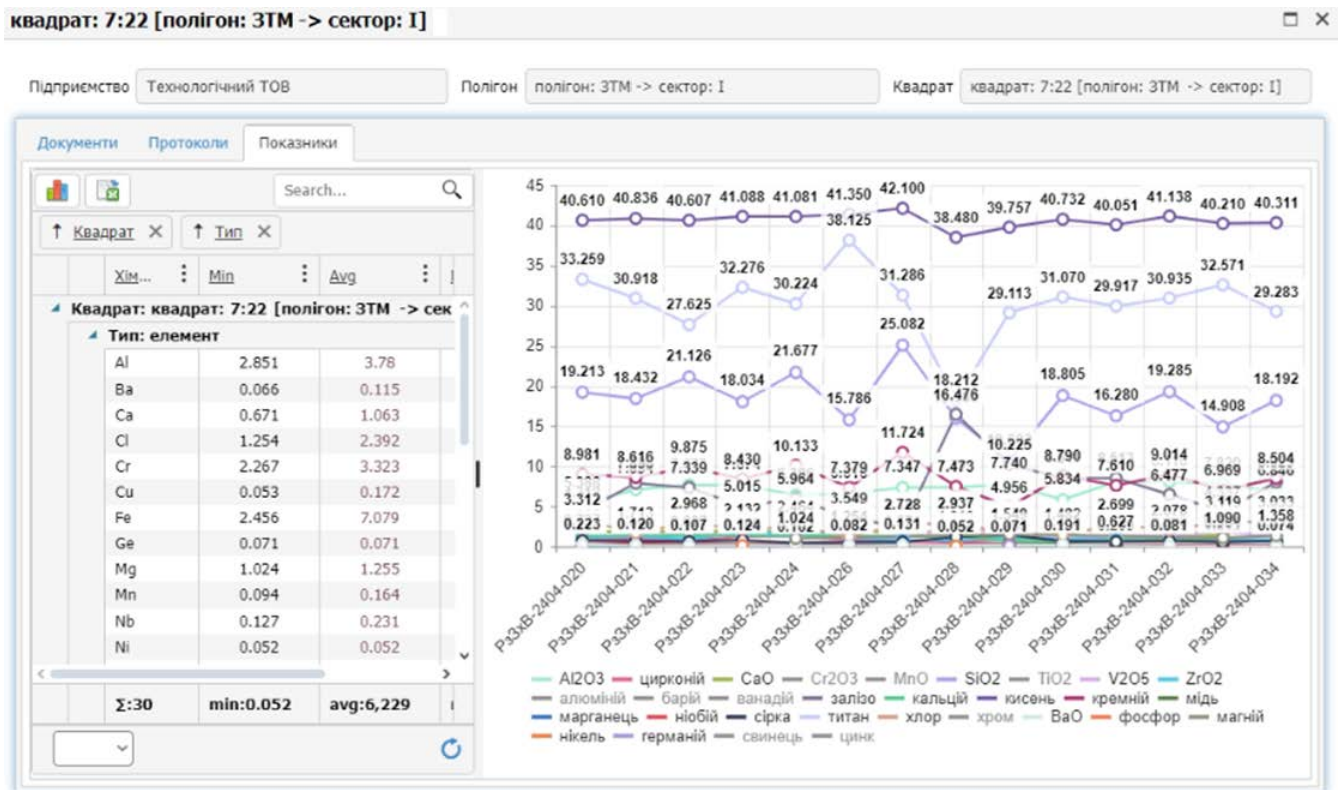


Рисунок 4.22 – Інтерфейс вікна перегляду результатів розвідки

За потреби дані можна вивантажити у формат Excel для подальшого аналізу та збереження. Це забезпечує додаткову гнучкість у роботі з інформацією та дозволяє легко інтегрувати результати розвідки з іншими системами чи документами.

Таким чином, можливість перегляду та експорту результатів у табличному та графічному форматі робить розроблений додаток ефективним інструментом для аналізу даних розвідки родовищ корисних копалин.

Лістинг коду усіх процесів наведено у Додатку В.

4.4 Тестування розробки та аналіз результатів

Метою тестування є перевірка працездатності та функціональності

розробленої автоматизованої системи для розвідки родовищ корисних копалин. Завдання тестування включають перевірку коректності обробки даних, стабільності системи під навантаженням, відповідності функціональних вимог та виявлення можливих помилок. У контексті розробки, тестування розуміється як процес визначення відповідності продукту початковим специфікаціям, які були задані технічним завданням [40].

Для тестування розробленої системи було обрано декілька методів, зокрема:

- Функціональне тестування – перевірка функціональності системи відповідно до специфікацій та вимог.
- Навантажувальне тестування – оцінка продуктивності системи під різними рівнями навантаження.
- Інтеграційне тестування – перевірка взаємодії між різними компонентами системи.
- Регресійне тестування – перевірка коректності роботи системи після внесення змін або виправлення помилок.

Функціональне тестування показало, що система виконує всі заявлені функції відповідно до вимог. Було проведено тестування додавання, редагування, видалення та оновлення даних, що підтвердило коректність обробки інформації.

Навантажувальне тестування продемонструвало стабільність системи під високими навантаженнями. Система змогла обробити великі обсяги даних без значних затримок та помилок.

Інтеграційне тестування підтвердило коректну взаємодію між різними модулями системи. Всі компоненти системи працювали злагоджено, забезпечуючи цілісність даних та їх коректну обробку.

Регресійне тестування показало, що внесені зміни та виправлення не вплинули на загальну функціональність системи. Всі виявлені помилки були успішно усунені, що підтверджується результатами контрольного тестування.

Для забезпечення високої якості та відповідності розробленої

автоматизованої системи для розвідки родовищ корисних копалин було складено таблицю 4.1 з критеріями приймання. Таблиця включає кілька ключових категорій, кожна з яких містить конкретні критерії, що повинні бути виконані перед впровадженням системи.

Виконання всіх критеріїв приймання забезпечить високу якість розробленої автоматизованої системи для розвідки родовищ корисних копалин та її готовність до повноцінного використання.

Таблиця 4.1 – Критерії приймання автоматизованої інформаційної системи для розвідки родовищ корисних копалин

Критерій	Опис	Результат
Функціональні вимоги		
Додавання даних	Можливість додавання нових записів у базу даних	позитивний
Редагування даних	Можливість редагування існуючих записів	позитивний
Видалення даних	Можливість видалення записів з бази даних	позитивний
Оновлення даних	Можливість оновлювати дані без втрати інформації	позитивний
Інтерфейс користувача		
Зручність використання	Інтерфейс є інтуїтивно зрозумілим та зручним для користувачів	позитивний
Доступність	Всі необхідні функції доступні з головного інтерфейсу	позитивний
Чіткість повідомлень	Система надає чіткі та інформативні повідомлення про помилки та успішні операції	позитивний
Обробка даних		
Коректність обробки	Всі операції з даними виконуються коректно	позитивний
Швидкість обробки	Система обробляє дані швидко та ефективно, навіть під високими навантаженнями	позитивний
Інтеграція даних	Дані з різних джерел інтегруються правильно	позитивний
Безпека		
Аутентифікація	Система вимагає аутентифікацію для доступу до інформації	позитивний
Захист даних	Дані захищені від несанкціонованого доступу та змін	позитивний
Звітування		
Формування звітів	Система може формувати звіти у різних форматах (табличний, графічний)	позитивний
Експорт у Excel	Дані можна експортувати у формат Excel	позитивний
Відображення даних	Всі необхідні дані відображаються коректно у звітах	позитивний

Продовження таблиці 4.1

Критерій	Опис	Результат
Тестування		
Функціональне тестування	Проведено успішне функціональне тестування	позитивний
Навантажувальне тестування	Проведено успішне навантажувальне тестування	позитивний
Інтеграційне тестування	Проведено успішне інтеграційне тестування	позитивний
Регресійне тестування	Проведено успішне регресійне тестування	позитивний
Відгуки користувачів		
Пілотне використання	Відгуки від пілотних користувачів про роботу системи	позитивний
Виправлення помилок	Внесення необхідні корективи на основі отриманих відгуків	позитивний

Розроблена автоматизована система для розвідки родовищ корисних копалин успішно пройшла всі етапи тестування. Система відповідає функціональним вимогам, демонструє високу продуктивність і стабільність під навантаженням, а також забезпечує коректну обробку даних. Виявлені помилки були виправлені, що підтверджується результатами регресійного тестування. Це свідчить про готовність системи до впровадження та використання в реальних умовах.

4.5 Висновки до розділу 4

Було детально розглянуто процес реалізації, опису та тестування інформаційної системи для розвідки родовищ корисних копалин. На основі проведеного аналізу та розробки можна зробити наступні висновки:

Реалізація інформаційної системи включала кілька ключових етапів, серед яких розробка фронтенд і бекенд частин, створення та налаштування бази даних, а також інтеграція різних компонентів для забезпечення їхньої злагодженої роботи. Використання сучасних веб-технологій і платформи Oracle Application Express (APEX) дозволило створити ефективну та гнучку систему.

Фронтенд частина системи забезпечує зручний інтерфейс для користувачів, що включає пошукову область із фільтрами, область кнопок управління й табличну область для відображення даних. Використання технологій HTML, CSS, JavaScript і бібліотеки Kendo UI дозволило створити інтерактивний та естетично привабливий інтерфейс.

Бекенд частина системи реалізована за допомогою мови програмування PL/SQL, що забезпечує високопродуктивну обробку даних у базі даних Oracle. Алгоритми обробки даних включають процедури для додавання, редагування, видалення та оновлення даних, що забезпечує актуальність і точність інформації в системі.

Інтеграція клієнтської та серверної частин системи здійснена за допомогою Oracle APEX, що дозволяє забезпечити тісну взаємодію між інтерфейсом користувача та серверною логікою. Це забезпечує оперативний обмін даними та підвищує зручність використання системи.

Тестування системи включало функціональне, навантажувальне, інтеграційне та регресійне тестування. Результати тестування підтвердили, що система відповідає заявленим функціональним вимогам, демонструє високу продуктивність і стабільність під навантаженням, а також забезпечує коректну обробку даних.

Автоматизована система для розвідки родовищ корисних копалин має значний потенціал для впровадження та використання в реальних умовах, оскільки дозволяє ефективно зберігати, каталогізувати та аналізувати геологічну інформацію, забезпечуючи швидкий доступ до необхідних даних для прийняття обґрунтованих рішень.

5 ЕКОНОМІЧНИЙ РОЗДІЛ

5.1 Технологічний аудит розробленої автоматизованої системи для розвідки родовищ корисних копалин

Як було підкреслено раніше, стрімкий технологічний прогрес суттєво трансформував геологорозвідувальну діяльність, відкривши нові горизонти для пошуку та аналізу родовищ корисних копалин. Завдяки цифровим технологіям процеси розвідки стали більш ефективними, точними і менш затратними, що має значний вплив на весь ланцюжок взаємовідносин між підприємствами гірничодобувної промисловості.

Цифровізація та автоматизація процесів розвідки відкрили шлях до створення інтегрованих систем управління даними, що дозволило ефективно зберігати, каталогізувати та аналізувати геологічну інформацію. Сучасні інформаційні системи допомагають об'єднати різноманітні дані, включаючи геологічні карти, результати буріння, лабораторні аналізи та інші, забезпечуючи швидкий доступ до необхідної інформації для прийняття обґрунтованих рішень.

Інтеграція цифрових технологій з традиційними методами геологічних досліджень створює нову парадигму в розвідці корисних копалин, знижує витрати та збільшує шанси на успішне виявлення та експлуатацію родовищ.

У виконаній нами магістерській кваліфікаційній роботі були продовжені дослідження в цьому напрямі. Зокрема, нами була розроблена автоматизована система для розвідки родовищ корисних копалин. Використання цієї системи для розвідки є не просто актуальним напрямком сталого розвитку, а й необхідною умовою для забезпечення технологічної й економічної ефективності, безпеки та екологічної відповідальності в галузі.

Для встановлення комерційного потенціалу розробленої нами автоматизованої системи для розвідки родовищ корисних копалин було запрошено трьох відомих та шанованих експертів – кандидата технічних наук, доцента Іванова Ю.Ю., кандидата технічних наук, доцента Гармаша В.В. та

кандидатку технічних наук, доценткиню Барабан М.В.

Встановлення комерційного потенціалу розробленої нами автоматизованої системи для розвідки родовищ корисних копалин було здійснено за критеріями, наведеними в таблиці 5.1.

Таблиця 5.1 – Рекомендовані критерії оцінювання комерційного потенціалу будь-якої розробки і їх бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту дещо гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
Ринкові перспективи					
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Запрошені експерти оцінили розроблену нами автоматизовану систему для розвідки родовищ корисних копалин таким чином (див. таблицю 5.2):

Таблиця 5.2 – Результати технологічного аудиту розробленої автоматизованої системи для розвідки родовищ корисних копалин (за шкалою оцінювання 0-1-2-3-4)

Критерії	Прізвище, ініціали експертів		
	Іванов Ю.Ю.	Гармаш В.В.	Барабан М.В.
	Бали, що їх виставили експерти:		
1	4	4	4
2	3	4	4
3	3	4	4
4	4	4	4
5	3	3	3
6	4	4	4
7	4	4	3
8	4	4	4
9	4	4	3
10	4	4	4
11	4	4	3
12	4	4	4
Сума балів	СБ ₁ = 45	СБ ₂ = 46	СБ ₃ = 44
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = \frac{45 + 46 + 44}{3} = \frac{135}{3} = 45,00$		

Встановлення комерційного потенціалу розробленої нами автоматизованої системи для розвідки родовищ корисних копалин будемо здійснювати на основі рекомендацій, наведених в таблиці 5.3 [41].

Таблиця 5.3 – Рівні комерційного потенціалу будь-якої наукової розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Оскільки середньоарифметична сума балів, що їх виставили експерти, складає 45 балів, то це свідчить, що розроблена нами автоматизована система для розвідки родовищ корисних копалин має рівень комерційного потенціалу, який вважається «високим».

Це пояснюється тим, що розроблена нами автоматизована система для розвідки родовищ корисних копалин базується на найсучасніших методах програмування і має невисоку вартість порівняно з іншими аналогічними системами. Запропонований нами інструментарій дозволить покращити процеси розвідки та видобутку корисних копалин, зменшити ризики та витрати при проведенні геологічних робіт, надасть експертам у галузі геології та розвідки нові можливості для аналізу та оцінки родовищ із значною економією часу та ресурсів.

Автоматизація процесів дозволить значно скоротити час на розвідку, знизити витрати на видобуток, а також підвищити точність і надійність отриманих даних. Це сприятиме зменшенню помилок, збільшенню обсягів ресурсів, що видобуваються, а також оптимізації використання природних ресурсів.

5.2 Розрахунок витрат на розроблення автоматизованої системи для розвідки родовищ корисних копалин

При розробці автоматизованої системи для розвідки родовищ корисних копалин нами були здійснені певні витрати.

Зокрема:

А). Основна заробітна плата Z_o розробників, яка визначається за формулою:

$$Z_o = \frac{M}{T_p} \cdot t \text{ грн}, \quad (5.1)$$

де M – місячний посадовий оклад розробника, грн; приймемо, що

$M = (8000...30000)$ грн/місяць;

T_p – число робочих днів в місяці; прийmemo $T_p = 20$ днів;

t – число днів роботи розробників.

Зроблені розрахунки зведемо до таблиці 5.4:

Таблиця 5.4 – Основна заробітна плата розробників

Найменування посади виконавця	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на оплату праці, грн
1. Науковий керівник магістерської роботи	30000	1500	20 годин	≈ 5000 (при 6-годинному робочому дні)
2. Магістрант-студент-виконавець	Беремо мінімальну оплату 8000 грн	400	85	≈ 34000
3. Консультант з економічної частини	20000	1000	1,5 години	≈ 250
4. Інші консультанти (з геологорозвідувальних робіт тощо)	40000	2000	2 дні	≈ 4000
Загалом				$З_o = 43250$ грн

Б). Додаткова заробітна плата $З_d$ розробників розраховується як $(10...12)\%$ від величини їх основної заробітної плати, тобто:

$$З_d = \alpha \cdot З_o = (0,1...0,12) \cdot З_o. \quad (5.2)$$

Прийmemo, що $\alpha = 0,1$. Тоді для нашого випадку отримаємо:

$$З_d = 0,1 \times 43250 = 4325 \text{ грн.}$$

В). Нарахування на заробітну плату $НЗП_{зп}$ розробників (дослідників) розраховуються за формулою:

$$\text{НЗП}_{\text{зп}} = (З_{\text{о}} + З_{\text{д}}) \cdot \frac{\beta}{100}, \quad (5.3)$$

де β – ставка обов'язкового єдиного внеску на державне соціальне страхування, %. $\beta = 22\%$. Тоді:

$$\text{НЗН}_{\text{зп}} = (43250 + 4325) \times 0,22 = 10466,50 \approx 10467 \text{ грн.}$$

Г). Амортизація основних засобів А, які використовувались під час виконання цієї роботи:

$$A = \frac{Ц \cdot Н_{\text{а}}}{100} \cdot \frac{T}{12} \text{ грн}, \quad (5.4)$$

де Ц – загальна балансова вартість основних засобів, грн;

На – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що $Н_{\text{а}} = (2,5 \dots 25)\%$;

T – термін використання основних засобів, місяці.

Зроблені розрахунки зведено в таблицю 5.5.

Таблиця 5.5 – Розрахунок амортизаційних відрахувань

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
1. Комп'ютерна техніка, обладнання, принтери тощо	72000	25,0	3,6 (при 80% використанні)	≈ 4320
2. Приміщення університету, кафедри	42000	2,5	3,6 при 60% використанні	≈ 189
Всього				$A = 4509 \text{ грн}$

Д). Витрати на матеріали М розраховуються за формулою:

$$M = \sum_{i=1}^n H_i \cdot Ц_i \cdot K_i - \sum_{i=1}^n B_i \cdot Ц_{\text{в}} \text{ грн.}, \quad (5.5)$$

де H_i – витрати матеріалу i -го найменування, кг; $Ц_i$ – вартість матеріалу i -го най-

менування; K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$; V_i – маса відходів матеріалу i -го найменування; Π_v – ціна відходів матеріалу i -го найменування; n – кількість видів матеріалів.

Е). Витрати на комплектуючі K розраховуються за формулою:

$$K = \sum_{i=1}^n H_i \cdot \Pi_i \cdot K_i \text{ грн}, \quad (5.6)$$

де H_i – кількість комплектуючих i -го виду, шт.; Π_i – ціна комплектуючих i -го виду; K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$; n – кількість видів комплектуючих.

Під час виконання магістерської кваліфікаційної роботи загальні витрати на матеріали та комплектуючі становили приблизно 2500 грн.

Ж). Витрати на силову електроенергію V_e розраховуються за формулою:

$$V_e = \frac{B \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_d},$$

(5.7)

де B – вартість 1 кВт-год. електроенергії, в 2024 р. $B \approx 4,5$ грн/кВт;

Π – установлена потужність обладнання, кВт; $\Pi = 1,35$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

Прийmemo, що $\Phi = 390$ годин;

K_{Π} – коефіцієнт використання потужності; $K_{\Pi} < 1 = 0,84$.

K_d – коефіцієнт корисної дії, $K_d = 0,79$.

Тоді витрати на силову електроенергію будуть дорівнювати:

$$V_e = \frac{B \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_d} = \frac{4,5 \cdot 1,35 \cdot 390 \cdot 0,84}{0,79} = 2519,20 \approx 2520 \text{ грн.}$$

И). Інші витрати $V_{\text{інш}}$ можна прийняти як $(50 \dots 300)\%$ від основної заробітної плати розробників, тобто:

$$V_{\text{інш}} = (0,5 \dots 3) \times 3_0. \quad (5.8)$$

Для нашого випадку отримаємо:

$$V_{\text{інш}} = 0,60 \times 43250 = 25950 \text{ грн.}$$

К). Сума всіх попередніх статей витрат складає витрати на виконання

магістерської роботи безпосередньо розробником-магістрантом – В.

$$B = 43250 + 4325 + 10467 + 4509 + 2500 + 2520 + 25950 = 93521 \text{ грн.}$$

Л). Загальні витрати на розробку автоматизованої системи для розвідки родовищ корисних копалин $B_{\text{заг}}$ розраховуються за формулою:

$$B_{\text{заг}} = \frac{B}{\beta}, \quad (5.9)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання цієї роботи.

Можна прийняти, що, $\beta \approx 0,84$ [41].

$$\text{Тоді: } B_{\text{заг}} = \frac{93521}{0,84} = 111334,52 \text{ грн або приблизно 112 тисяч грн.}$$

Тобто прогнозовані загальні витрати на розробку автоматизованої системи для розвідки родовищ корисних копалин становлять приблизно 112 тисяч грн.

5.3 Розрахунок економічного ефекту від можливої комерціалізації розробки

Економічний ефект від впровадження та можливої комерціалізації розробленої нами автоматизованої системи для розвідки родовищ корисних копалин пояснюється її значно кращими та ширшими функціональними можливостями. Тому нашу розробку можна реалізовувати на ринку дещо дорожче, ніж подібні (але значно гірші) за функціями розробки. Так, якщо подібна за функціями система може коштувати на ринку приблизно 130 тисяч грн, то нашу розробку можна буде реалізовувати на ринку приблизно за 150 тисяч грн або на 20 тисяч грн дорожче.

Аналіз ринку показує, що цільову аудиторію для нашої розробки можуть скласти геологи, геофізики, гірничі інженери, менеджери проєктів, аналітики та економісти тощо. А потенційними споживачами нашої розробки можуть бути гірничодобувні компанії; геологорозвідувальні компанії; науково-дослідні інститути та університети; консалтингові компанії тощо. Тобто попит на

інформаційні системи для розвідки родовищ корисних копалин є досить високим і постійно зростає та охоплює широкий спектр ринків.

Аналіз місткості ринку показує, що на сьогодні в Україні кількість потенційних споживачів розробленої нами автоматизованої системи для розвідки родовищ корисних копалин може становити приблизно 100 зацікавлених осіб. Аналіз ринку також показує, що можна очікувати зростання попиту на нашу розробку принаймні протягом 3-х років після її впровадження.

Тобто, якщо наша розробка буде впроваджена з 1 січня 2025 року, то її результати будуть виявлятися протягом 2025-го, 2026-го та 2027-го років.

Прогноз зростання попиту на нашу розробку складає по роках:

- а) 2025 р. – приблизно +20 шт. до базового року;
- б) 2026 р. – +30 шт. до базового року;
- в) 2027 р. – +40 шт. до базового року.

Можливе збільшення чистого прибутку $\Delta\Pi_i$, що його може отримати потенційний інвестор від комерціалізації нашої розробки, тобто виведення нашої розробки на ринок, становитиме:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{v}{100}\right), \quad (5.10)$$

де $\Delta\Pi_o$ – покращення основного якісного показника від впровадження результатів нашої розробки у цьому році. Для нашого випадку це є збільшення ціни реалізації нашої розробки $\Delta\Pi_o = (150 - 130) = +20$ тисяч грн;

N – основний кількісний показник, який визначає обсяг діяльності у році до впровадження результатів розробки; $N = 100$ шт.;

ΔN – покращення основного кількісного показника від впровадження результатів розробки. Таке покращення становитиме по роках, відповідно:

у 2025 році – +20 шт., у 2026 році +30 шт., та у 2027 році +40 шт. (відносно базового 2024 року);

Π_o – основний якісний показник (тобто ціна), який визначає обсяг діяльності

у році після впровадження результатів розробки, грн; $C_0 = 150$ тисяч грн;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки; для нашого випадку $n = 3$;

λ – коефіцієнт, який враховує сплату податку на додану вартість; $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати $\rho = (0,2 \dots 0,5)$; візьмемо $\rho = 0,4$;

ν – ставка податку на прибуток. У 2024 $\nu = 18\%$. У 2025 році і в наступних роках також очікуємо ставку $\nu = 18\%$.

Тоді можливе зростання чистого прибутку $\Delta \Pi_1$ для потенційного інвестора протягом першого року від можливого впровадження (комерціалізації) нашої розробки (2025 р.) складе:

$$\Delta \Pi_1 = [20 \cdot 100 + 150 \cdot 20] \cdot 0,8333 \cdot 0,4 \cdot \left(1 - \frac{18}{100}\right) \approx 1367 \text{ тис. грн.}$$

Можливе зростання чистого прибутку $\Delta \Pi_2$ для потенційного інвестора від можливого впровадження (комерціалізації) нашої розробки протягом другого (2026 р.) року складе:

$$\Delta \Pi_2 = [20 \cdot 100 + 150 \cdot 30] \cdot 0,8333 \cdot 0,4 \cdot \left(1 - \frac{18}{100}\right) \approx 1777 \text{ тис. грн.}$$

Можливе зростання чистого прибутку $\Delta \Pi_3$ для потенційного інвестора від можливого впровадження (комерціалізації) нашої розробки протягом третього (2027 р.) року становитиме:

$$\Delta \Pi_3 = [20 \cdot 100 + 150 \cdot 40] \cdot 0,8333 \cdot 0,4 \cdot \left(1 - \frac{18}{100}\right) \approx 2186 \text{ тис. грн.}$$

Приведена вартість зростання всіх чистих прибутків, що їх може отримати потенційний інвестор від можливого впровадження нашої розробки становитиме:

$$\Pi\Pi = \sum_{i=1}^T \frac{\Delta \Pi_i}{(1 + \tau)^t}, \quad (5.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої роботи, грн;

t – період часу, протягом якого виявляються результати впровадженої роботи, роки. Для нашого випадку $t = 3$ роки;

τ – ставка дисконтування. Прийmemo $\tau = 0,10$ (10%);

t – період часу від моменту початку розроблення автоматизованої системи для розвідки родовищ корисних копалин до моменту отримання можливих чистих прибутків потенційним інвестором.

Тоді приведена вартість зростання всіх можливих чистих прибутків $\Pi\Pi^{10\%}$ (при 10% інфляції), що їх може отримати потенційний інвестор від комерціалізації нашої розробки, складе:

$$\Pi\Pi^{10\%} = \frac{1367}{(1+0,1)^2} + \frac{1777}{(1+0,1)^3} + \frac{2186}{(1+0,1)^4} \approx 1130 + 1335 + 1493 = 3958 \text{ тисяч грн.}$$

Теперішня вартість інвестицій PV , що можуть бути вкладені для реалізації нашої розробки: $PV = K \times B_{\text{заг}} = (1,0 \dots 5) \times B_{\text{заг}}$.

Прийнявши $K = 4$, для нашого випадку прийmemo:

$$PV = (1,0 \dots 5) \times 112 = 4,0 \times 112 = 448 \text{ тисяч грн.}$$

Абсолютний ефект від можливих вкладених інвестицій $E_{\text{абс}}$ становитиме:

$$E_{\text{абс}} = \Pi\Pi - PV, \quad (5.12)$$

де $\Pi\Pi$ – приведена вартість збільшення всіх чистих прибутків для потенційного інвестора від можливого впровадження нашої розробки, грн;

PV – теперішня вартість інвестицій $PV = 448$ тисяч грн.

Тобто абсолютний ефект від можливого впровадження нашої розробки (при прогнозованому ринку збуту) за три роки складе:

$$E_{\text{абс}} = 3958 - 448 = 3510 \text{ тисяч грн.}$$

Оскільки $E_{\text{абс}} > 0$, то комерціалізація нашої розробки може бути доцільною.

Далі розрахуємо внутрішню дохідність E_v вкладених інвестицій:

$$E_B = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (5.13)$$

де $E_{\text{абс}}$ – абсолютний ефект вкладених інвестицій; $E_{\text{абс}} = 3510$ тис. грн;

PV – теперішня вартість початкових інвестицій $PV = 448$ тис. грн;

$T_{\text{ж}}$ – життєвий цикл розробки, роки.

$T_{\text{ж}} = 4$ років (2024-й, 2025-й, 2026-й, 2027-й роки)

Для нашого випадку отримаємо:

$$E_B = \sqrt[4]{1 + \frac{3510}{448}} - 1 = \sqrt[4]{1 + 7,8348} - 1 = \sqrt[4]{8,8348} - 1 \approx 1,72 - 1 = 0,72 = 72,0\%.$$

Далі визначимо ту мінімальну дохідність, нижче за яку потенційному інвестору не вигідно буде займатися комерціалізацією нашої розробки.

Мінімальна дохідність або мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою: $\tau_{\text{мін}} = d + f$, (5.14)

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = (0,14...0,18)$;

f – показник, що характеризує ризикованість вкладень; $f = (0,05...0,40)$.

Для нашого випадку отримаємо:

$$\tau_{\text{мін}} = 0,15 + 0,35 = 0,50 \text{ або } \tau_{\text{мін}} = 50\%.$$

Оскільки величина $E_B = 72,0\% > \tau_{\text{мін}} = 50\%$, то потенційний інвестор у принципі може бути зацікавлений у фінансуванні та комерціалізації розробленої нами автоматизованої системи для розвідки родовищ корисних копалин.

Далі розраховуємо термін окупності коштів, вкладених у можливу комерціалізацію розробленої нами автоматизованої системи для розвідки родовищ корисних копалин.

Термін окупності $T_{\text{ок}}$ розраховується за формулою:

$$T_{\text{ок}} = \frac{1}{E_B}. \quad (5.15)$$

Для нашого випадку термін окупності $T_{\text{ок}}$ коштів становитиме:

$$T_{\text{ок}} = \frac{1}{0,72} = 1,39 \text{ років} < 3 \text{ років},$$

що свідчить про потенційну доцільність комерціалізації розробленої нами автоматизованої системи для розвідки родовищ корисних копалин.

Далі проведено моделювання залежності величини внутрішньої дохідності вкладених інвестором потенційних інвестицій від рівня інфляції в країні (який в умовах війни може зростати).

Прийнявши рівень інфляції у 20%, отримаємо:

$$\text{ПП}^{20\%} = \frac{1367}{(1+0,2)^2} + \frac{1777}{(1+0,2)^3} + \frac{2186}{(1+0,2)^4} \approx 949 + 1028 + 1054 = 3031 \text{ тисяч грн.}$$

Тоді абсолютний ефект від можливого впровадження нашої розробки за три роки складе:

$$E_{\text{абс}} = 3031 - 448 = 2583 \text{ тисяч грн.}$$

Внутрішня дохідність E_v вкладених інвестицій становитиме:

$$E_v = T_{\text{ж}} \sqrt[4]{1 + \frac{E_{\text{абс}}}{PV}} - 1,$$

де $E_{\text{абс}}$ – абсолютний ефект вкладених інвестицій; $E_{\text{абс}} = 2583$ тисяч грн;

PV –теперішня вартість початкових інвестицій $PV = 448$ тисяч грн.

Для нашого випадку отримаємо:

$$E_v = \sqrt[4]{1 + \frac{2583}{448}} - 1 = \sqrt[4]{1 + 5,7656} - 1 = \sqrt[4]{6,7656} - 1 \approx 1,61 - 1 = 0,61 = 60,1\%.$$

Прийнявши рівень інфляції у 30%, отримаємо:

$$\text{ПП}^{30\%} = \frac{1367}{(1+0,3)^2} + \frac{1777}{(1+0,3)^3} + \frac{2186}{(1+0,3)^4} \approx 809 + 809 + 766 = 2384 \text{ тисяч грн.}$$

Тоді абсолютний ефект від можливого впровадження нашої розробки за три роки складе:

$$E_{\text{абс}} = 2384 - 448 = 1936 \text{ тисяч грн.}$$

Внутрішня дохідність E_v вкладених інвестицій становитиме:

$$E_v = T_{\text{ж}} \sqrt[4]{1 + \frac{E_{\text{абс}}}{PV}} - 1,$$

де $E_{\text{абс}}$ – абсолютний ефект вкладених інвестицій; $E_{\text{абс}} = 1936$ тисяч грн;

PV –теперішня вартість початкових інвестицій $PV = 448$ тисяч грн.

Для нашого випадку отримаємо:

$$E_b = \sqrt[4]{1 + \frac{1936}{448}} - 1 = \sqrt[4]{1 + 4,3214} - 1 = \sqrt[4]{5,3214} - 1 \approx 1,518 - 1 = 0,518 = 51,8\%.$$

Зроблені розрахунки у вигляді графіків наведено на рисунку 5.1.

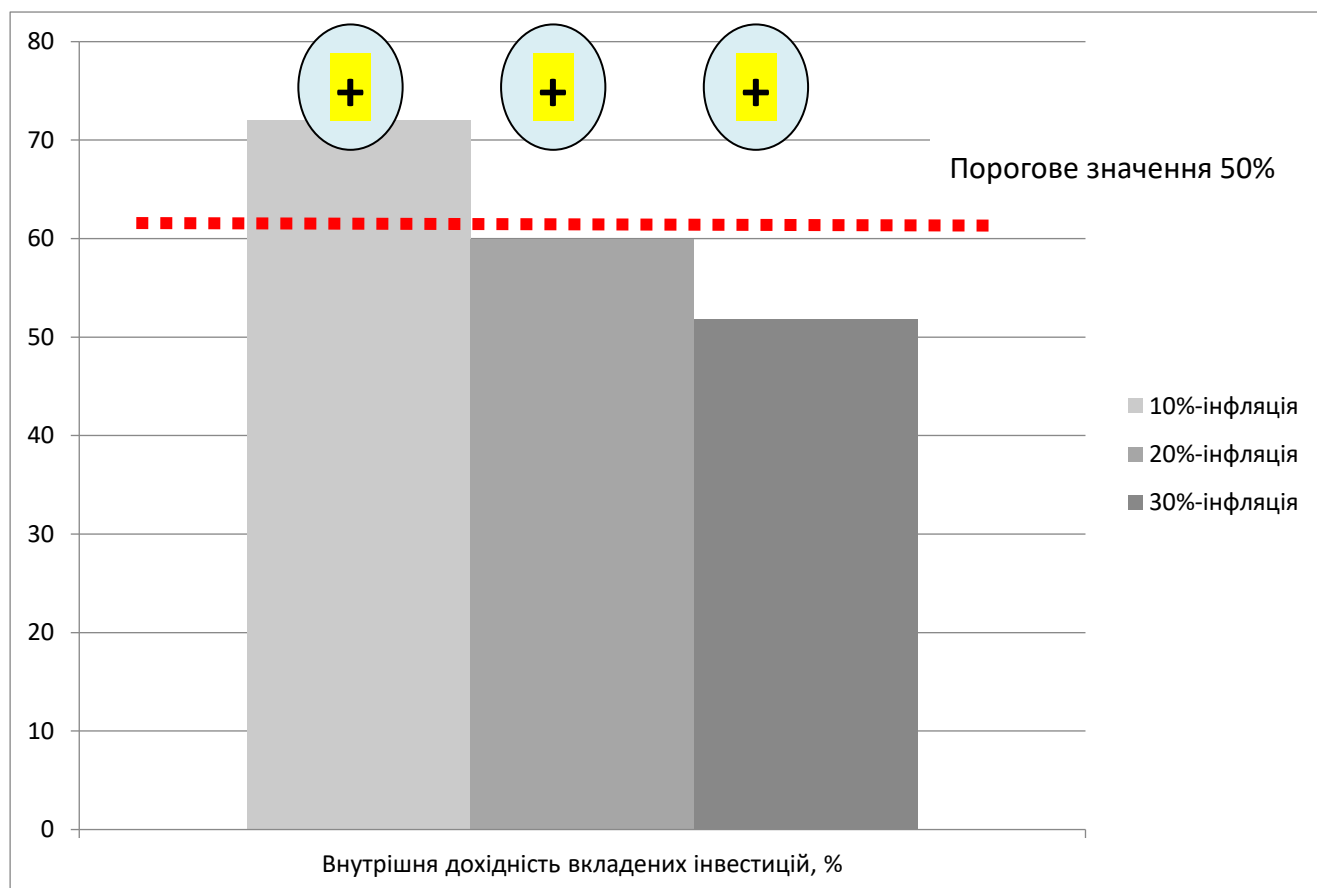


Рисунок 5.1 – Моделювання залежності величини внутрішньої дохідності потенційних інвестицій, вкладених в комерціалізацію розробленої автоматизованої системи для розвідки родовищ корисних копалин, від рівня інфляції в країні

Аналіз графіку на рисунку 5.1 показує, що при рівні інфляції в країні до 30% величина внутрішньої дохідності інвестицій буде перевищувати порогове значення $\tau_{\min} = 50\%$, і тому комерціалізація нашої розробки може бути доцільною.

Результати виконаної економічної частини магістерської кваліфікаційної роботи зведено у таблицю 5.6:

Таблиця 5.6 – Результати виконаної економічної частини магістерської кваліфікаційної роботи

Показники	Задані у ТЗ	Досягнуті у магістерській кваліфікаційній роботі	Висновок
1. Витрати на розробку	Не більше 120 тисяч грн	112 тисяч грн.	Досягнуто
2. Абсолютний ефект від впровадження (комерціалізації) розробки, тисяч грн	Не менше 3500 тисяч грн (за три роки)	3510 тисяч грн	Виконано
3. Внутрішня дохідність потенційних інвестицій, %	не менше 50%	72,0%	Досягнуто
4. Термін окупності потенційних інвестицій, роки	до 3-ти років	1,39 років	Виконано

Таким чином, основні техніко-економічні показники розробленої нами автоматизованої системи для розвідки родовищ корисних копалин, визначені у технічному завданні, виконані.

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи було розроблено автоматизовану інформаційну систему для розвідки родовищ корисних копалин, яка значно підвищує ефективність і точність геологорозвідувальних робіт. Розроблена система дозволяє оптимізувати процеси збору, зберігання, обробки та аналізу геологічних даних, забезпечуючи швидкий доступ до інформації і підтримку прийняття обґрунтованих рішень.

У першому розділі роботи було проведено аналіз існуючих методів та засобів автоматизації геологорозвідувальних робіт. Визначено основні проблеми і вимоги, які стоять перед сучасними інформаційними системами для розвідки родовищ корисних копалин. Аналіз показав, що більшість існуючих рішень не забезпечують достатнього рівня інтеграції даних та автоматизації процесів. Це зумовило необхідність розробки нової системи, яка б відповідала сучасним стандартам і була б здатна ефективно обробляти великі обсяги геологічної інформації.

Другий розділ роботи присвячено розробці структури автоматизованої інформаційної системи. Було описано архітектуру системи, обрано відповідні технології та інструменти для реалізації кожного з її компонентів. Зокрема, було обрано Oracle APEX для розробки інтерфейсу користувача та PL/SQL для реалізації серверної логіки. Описано інтеграцію з іншими програмними продуктами, що забезпечує гнучкість і розширюваність системи. В результаті розроблено детальну архітектуру системи, яка включає всі необхідні компоненти для ефективної роботи з геологічними даними.

Третій розділ роботи зосереджено на розробці алгоритмів для роботи програмного продукту. Було розроблено алгоритми, які забезпечують ефективну обробку даних, їх збереження та управління у базі даних. Ці алгоритми включають процедури для додавання, редагування, видалення та оновлення даних. Проведено тестування алгоритмів на реальних даних, що підтвердило їх ефективність та надійність. Особливу увагу приділено забезпеченню високої продуктивності системи та оптимізації процесів обробки даних.

У четвертому розділі детально описано процес реалізації системи, включаючи фронтенд та бекенд компоненти, а також алгоритми обробки даних у базі даних. Розглянуто етапи розробки користувацького інтерфейсу, описано функціональні можливості системи та проведено її тестування на реальних даних. Описано процес створення та налаштування бази даних, розробку інтерфейсів для введення та редагування даних, а також реалізацію основних функцій системи. Проведено тестування системи, яке показало її високу продуктивність і стабільність під навантаженням, а також підтвердило коректність обробки даних.

П'ятий розділ присвячено економічному обґрунтуванню впровадження розробленої системи. Було проведено аналіз витрат на розробку та впровадження системи, а також оцінено потенційні економічні вигоди від її використання. Встановлено, що впровадження автоматизованої системи дозволяє значно знизити витрати на геологорозвідувальні роботи та підвищити їх ефективність. Було розраховано економічний ефект від впровадження системи, який включає зниження операційних витрат, підвищення продуктивності праці та покращення якості прийнятих рішень. Проведені розрахунки доводять, що система може мати абсолютний ефект від впровадження розміром 3510 тисяч грн за три роки. В результаті було доведено, що впровадження системи є економічно доцільним та сприяє сталому розвитку підприємств гірничодобувної галузі.

На основі проведених досліджень можна зробити висновок, що розроблена автоматизована інформаційна система для розвідки родовищ корисних копалин є ефективним інструментом для сучасних геологорозвідувальних підприємств. Вона дозволяє значно підвищити точність і швидкість обробки геологічних даних, знизити витрати на проведення розвідувальних робіт, а також забезпечити більш обґрунтоване прийняття рішень щодо експлуатації родовищ. Розроблена система відповідає сучасним стандартам та вимогам, демонструє високу продуктивність та стабільність у роботі, що підтверджується результатами тестування та пілотного впровадження. Впровадження системи сприятиме підвищенню ефективності геологорозвідувальних робіт, зниженню ризиків та покращенню економічних показників підприємств гірничодобувної галузі,

зменшенню ризиків для навколишнього середовища та загальному підвищенню екологічної відповідальності підприємств, що займаються видобутком корисних копалин.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мазуренко Л.Л., Черненко О.В. Цифрові технології в геологорозвідці: стан та перспективи. Наукові праці ВНТУ. 2018. Вип. 2. С. 112–119.
2. Юрчук М.С., Паламарчук Є.А. Автоматизовані інформаційні системи як засіб підвищення ефективності розвідки родовищ корисних копалин. Науково-технічна конференція факультету комп'ютерних систем і автоматики (2024). матеріали наук.-практ. конф., м. Вінниця. Вінниця, 2024.
3. Басок Б.І., Тимчук І.Г., Романенко В.Д. Екологічні аспекти використання цифрових технологій у гірничодобувній промисловості. Екологічний вісник. 2021. № 4. С. 76–83.
4. Інноваційні підходи в розвідці родовищ корисних копалин. Звіт Національної гірничої академії України. 2020.
5. Лісовий І.М., Петровський О.З. Сучасні підходи до розвідки корисних копалин з використанням ГІС технологій. Геологічний журнал. 2020. № 3. С. 88–97.
6. Lillesand, T. M., Kiefer, R. W., & Chipman, J. Remote Sensing and Image Interpretation. John Wiley & Sons, 2015.
7. Петренко М.Д., Сидоренко В.Г. Вплив штучного інтелекту на ефективність розвідки родовищ. Вісник Вінницького політехнічного інституту. 2017. № 5. С. 134–142.
8. S&P Global Commodity Insights. Ukraine may advance with critical minerals project auctions this year: Geological Service. URL: <https://www.spglobal.com/commodityinsights/en/market-insights/latest-news/metals/051021-ukraine-may-advance-with-critical-minerals-project-auctions-this-year-geological-service> (дата звернення: 06.03.2024).
9. Martin, R., Muuls, M. The Impact of Green Technology on Economic Development. Journal of Sustainable Development, 2020. Vol. 13, No. 4. pp. 28-42.
10. How automation in Australia could be the future of mining. URL:

<https://www.mining-technology.com/features/how-automation-in-australia-could-be-the-future-of-mining/> (дата звернення: 08.04.2024).

11. Лисенко В.І. Геологічна розвідка та розробка корисних копалин. Київ: Наукова думка, 2016.
12. Георгієв Г.Ю., Ковальчук М.П., Шевчук С.А. Застосування інформаційних систем в геологічних дослідженнях та розвідці корисних копалин. Науковий вісник НГУ. 2019. № 6. С. 45–53.
13. Steiniger, S., Bocher E. An Overview on Current Free and Open Source Desktop GIS Developments. International Journal of Geographical Information Science. 2009. 23(10). pp. 1345-1370.
14. MapInfo Professional User Guide. Pitney Bowes Software, 2019.
15. ENVI User's Guide. L3Harris Geospatial Solutions, 2021.
16. Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. Google Earth Engine: Planetary-scale geospatial analysis for everyone. Remote Sensing of Environment. 2017. 202. pp. 18-27.
17. Petrel E&P Software Platform. Schlumberger. 2021.
18. Brown, A. Practical Database Programming with SQL Server. Wiley. 2016.
19. Duckett J. HTML and CSS: Design and Build Websites. Wiley. 2011.
20. Crockford D. JavaScript: The Good Parts. O'Reilly Media. 2008.
21. Keating T. CSS Master (2nd ed.). Apress. 2020.
22. Progress Kendo UI. (n.d.). Kendo UI Documentation. Progress Software Corporation. URL: docs.telerik.com/kendo-ui (дата звернення: 05.04.2024).
23. White J., Simpson E. Professional Kendo UI: Building Rich Internet Applications with HTML5 and JavaScript. Wrox. 2014.
24. Stuttard D., Pinto, M. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws (2nd ed.). Wiley. 2017.
25. Choi M. Web Security for Developers: Real Threats, Practical Defense. No Starch Press. 2020
26. Oracle Application Express (APEX). Oracle APEX Documentation.

Oracle Corporation, n.d., URL: docs.oracle.com/en/database/oracle/application-express/index.html (дата звернення: 05.04.2024).

27. Scott J., Scott S. Pro Oracle Application Express. Apress. 2016
28. Feuerstein S., Bill P. Oracle PL/SQL Programming. 6th ed. O'Reilly Media, 2014.
29. Oracle Database PL/SQL Language Reference. (n.d.). Oracle Documentation. Oracle Corporation. URL: <https://docs.oracle.com/en/database/oracle/oracle-database/19/lnpls/index.html> (дата звернення: 10.04.2024).
30. Todd A. Mastering Markdown. Leanpub. 2017.
31. Desjardins R. Learning TypeScript 2.x: Develop and maintain captivating web applications with ease. Packt Publishing. 2020.
32. Banks A., Porcello E. Learning React: Functional Web Development with React and Redux. 2nd ed. O'Reilly Media. 2020
33. Rohde A., Farrelly N. Learning Angular: A Guide to Modern Web Development. O'Reilly Media. 2018
34. Kuehl H., Deutsch F. Fullstack Vue: The Complete Guide to Vue.js. Fullstack.io. 2017.
35. Марченко А. В. Проектування інформаційних систем. 2016.
36. Krug S. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. 3rd ed. New Riders. 2014.
37. Tidwell J. Designing Interfaces. 3rd ed. O'Reilly Media. 2019.
38. Models of Database Architecture: Hierarchical, Network and Relational Models. YourArticleLibrary. URL: <https://www.yourarticlelibrary.com/database/models-of-database-architecture-hierarchical-network-and-relational-models/> (дата звернення: 10.04.2024)
39. Garcia-Molina H., Ullman J. D., Widom, J. Database Systems: The Complete Book 2nd ed. Pearson. 2020.
40. Авраменко А.С., Авраменко В.С., Косенюк Г.В. Тестування програмного забезпечення. Навчальний посібник. Черкаси: ЧНУ імені Богдана

Хмельницького. 2017. 284 с

41. Козловський В.О., Лесько О.Й., Кавецький В.В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. Вінниця: ВНТУ. 2021. 42 с.

ДОДАТКИ

Додаток А (обов'язковий)

Технічне завдання на магістерську кваліфікаційну роботу

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ

____ д.т.н., проф. Олег БІСІКАЛО

«__» _____ 2024 року

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу

Автоматизована інформаційна система для розвідки родовищ корисних копалин

08-31.МКР.007.02.000 ТЗ

Керівник: професор кафедри АІТ ВНТУ,
к.т.н., доцент

_____ Євген ПАЛАМАРЧУК

«__» _____ 2024 р.

Виконавець: студент гр. АКІТ-22мз

_____ Марія ЮРЧУК

«__» _____ 2024 р.

1. Назва та галузь застосування.

Автоматизована інформаційна система для розвідки родовищ корисних копалин. Автоматизація та комп'ютерно-інтегровані технології.

2. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____2024р., та індивідуальне завдання на МКР, затверджене протоколом №__ засідання кафедри АІТ від «__» _____ 2024р.

3. Мета та призначення роботи.

Метою магістерської дипломної роботи є розробка автоматизованої інформаційної системи для розвідки родовищ корисних копалин.

4. Джерела розробки:

4.1 Методичні вказівки до виконання магістерських дипломних робіт (проектів) для студентів спеціальностей 126 – «Інформаційні системи та технології», 151 – «Автоматизація та комп'ютерно-інтегровані технології» / Уклад. Р. Н. Кветний, О. М. Бевз, О. В. Бісікало. Вінниця : ВНТУ, 2019. 26 с.

4.2 ДСТУ 2481-94 Системи оброблення інформації. Інтелектуальні інформаційні технології. Терміни та визначення.

4.3 ДСТУ ISO/IEC/IEEE 16326:2015 Розроблення систем та програмного забезпечення. Процеси життєвого циклу. Керування проектами (ISO/IEC/IEEE 16326:2009, IDT).

4.4 ДСТУ ISO/IEC 12207:2016 Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення (ISO/IEC 12207:2008, IDT).

4.5. ДСТУ 3330-96 (ГОСТ 34.321-96) Інформаційні технології. Система стандартів з баз даних. Еталонна модель керування даними

5. Показники призначення

Основні технічні вимоги та мінімальні системні вимоги до програми:

5.1 Мови програмування – JavaScript

5.2 Оперативна пам'ять – 2 GB

5.3 Вільного місця – 1 GB

5.4 Бібліотека для створення інтерфейсу користувача – JQuery, Kendo Ui

5.5 Галузь використання – комерційна

5.6 Браузер – Google Chrome

5.7 Середовище розробки – Visual Studio Code

Вихідні дані для проведення робіт:

Вихідними даними для проведення робіт є індивідуальне завдання на магістерську дипломну роботу від _____ 2024 р.

Результати роботи програми: розроблено програмне забезпечення для автоматизованої інформаційної системи для розвідки родовищ корисних копалин. Описано архітектуру клієнт-серверної системи із використанням HTML+CSS, JavaScript у поєднанні з бібліотекою JQuery, а також Oracle Application Express, який постачається разом з ліцензією сервера Oracle Database. Розроблена інформаційна система реалізована та успішно протестована.

6. Економічні показники

До економічних показників входять:

- витрати на розробку 112 тисячі гривень.
- приведена вартість прибутку за 3 роки 3510 тисяч гривень
- внутрішня дохідність 2384 тисяч гривень при 30% інфляції
- термін окупності 1,39 років

7. Стадії розробки:

- а) Аналіз предметної області та визначення аналогів програмних продуктів —
- б) Вибір технологічних засобів та розробка архітектури —
- в) Обґрунтування та проєктування програмного забезпечення —
- г) Розробка технічного завдання —
- д) Економічна частина —
- е) Аналіз результатів роботи та формування завдань подальших досліджень —
- ж) Оформлення матеріалів до захисту МКР —

8. Порядок контролю та приймання

Рубіжний контроль провести до «___» _____ 2024 р.

Попередній захист МКР провести «___» _____ 2024 р.

Захист МКР провести до «___» _____ 2024 р.

Розробив студент групи АКІТ-22мз _____ Марія ЮРЧУК

Додаток Б (обов'язковий)

Ілюстративна частина

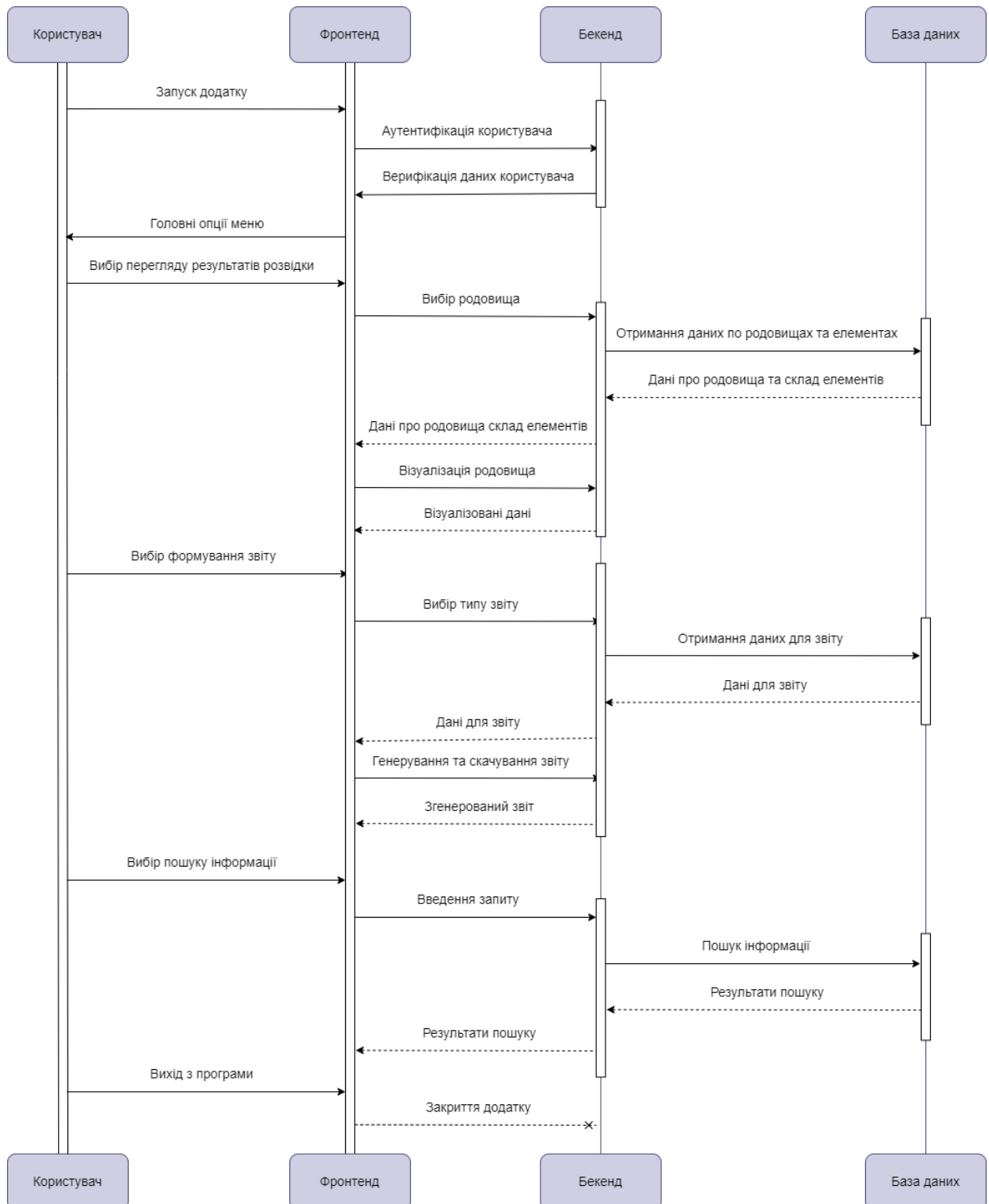


Рисунок Б.1 – Діаграма послідовностей автоматизованої інформаційної системи
для розвідки родовищ корисних копалин

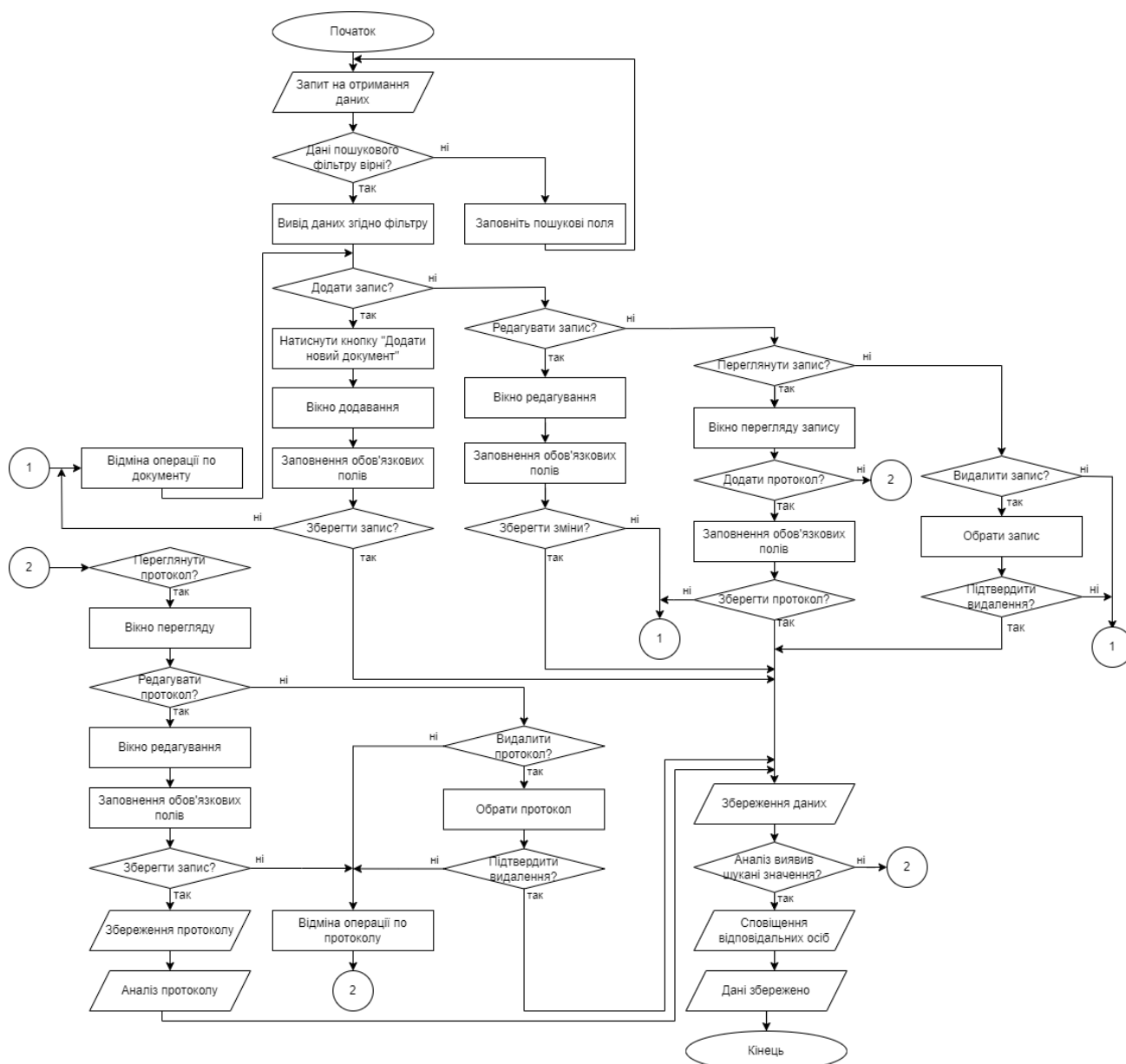


Рисунок Б.2 – Блок-схема функціонування інформаційної системи розвідки родовищ корисних копалин

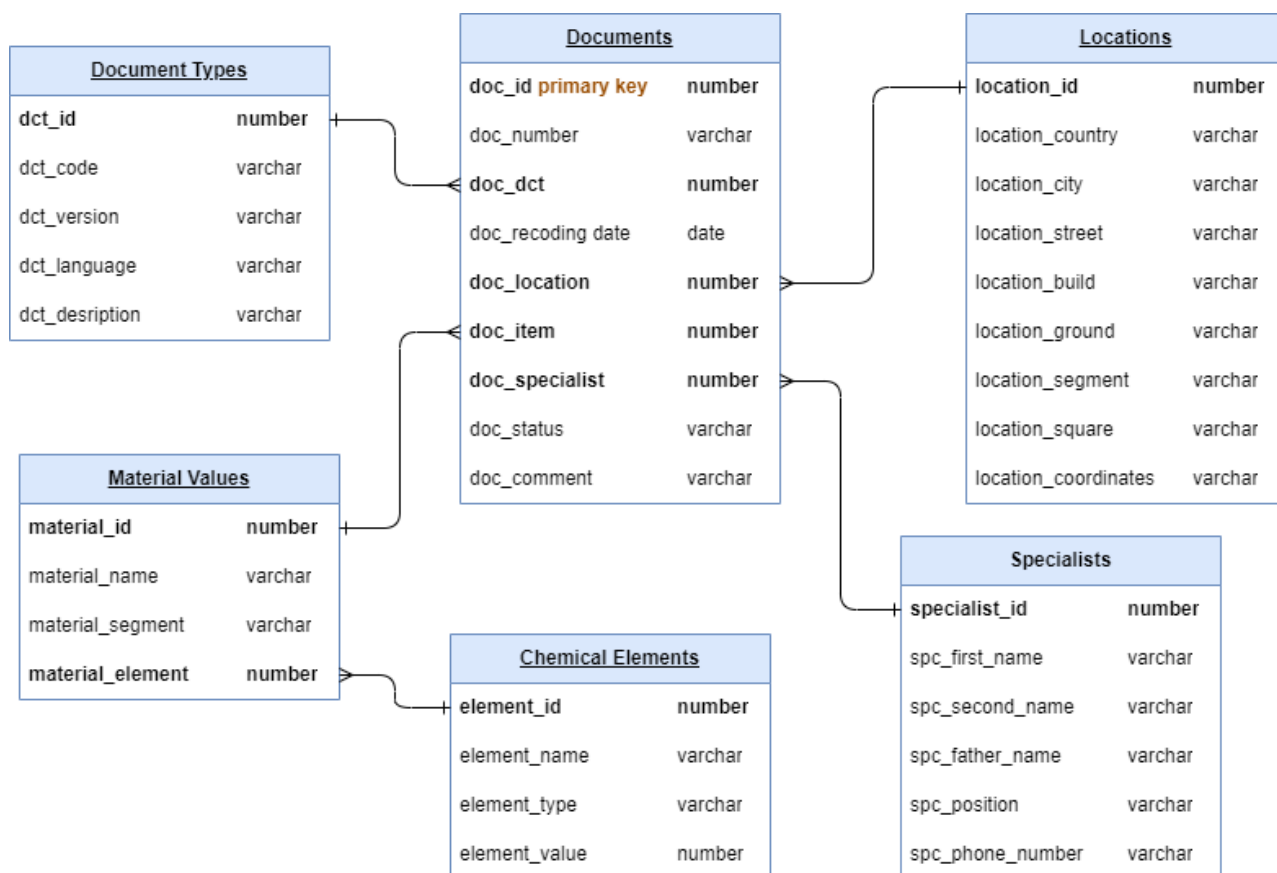


Рисунок Б.3 – Структура бази даних для інформаційної системи розвідки родовищ корисних копалин

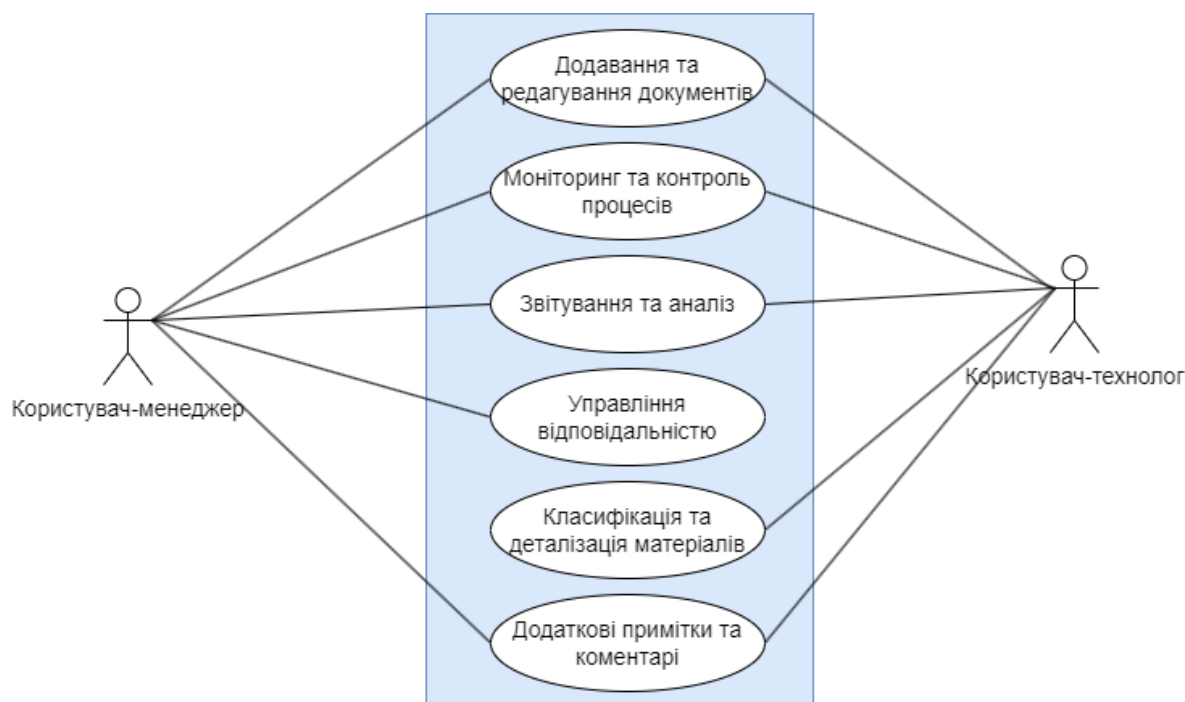


Рисунок Б.4 – Діаграма варіантів використання функціонального елемента «Полігони»



Рисунок Б.5 – Діаграма варіантів використання функціонального елемента «Документи»

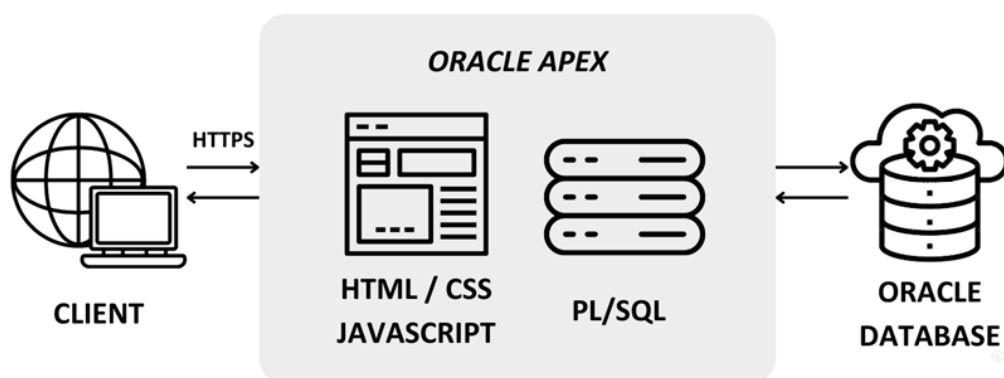


Рисунок Б.6 – Архітектура взаємодії між клієнтом, Oracle APEX та базою даних Oracle

Додаток В (обов'язковий)

Лістинг програми

```
// file 'main.sql'
select doc.*, pltec_common.GetDprFullName(p_dpr_id => doc.doc_dpr, p_type => 2) as dpr_name_full,
       pltec_common.GetSpcName2Date(p_spc => dspc.dspc_spc, p_type => 2) as doc_spc,
       dst.dic_sname as doc_st_sname, lower(doc.doc_st) as doc_st_low,
       dct.dct_name, dct.dct_code, dis_pltec.pltec_common.GetSpcName2Date(p_spc => dspc.dspc_spc) || '' ||
       (select oct_phone_main_1_mask from dis_pltec.vm_spc where spc_id = dspc.dspc_spc) || " as doc_spc_name_title,
       case when v('P1000_LANG') in ('UKR') then ist.ist_name_ukr_f else ist.ist_name_rus_f end as ist_place,
       case when v('P1000_LANG') in ('UKR') then ist.ist_name_par_ukr_f else ist.ist_name_par_rus_f end as ist_par,
       case when v('P1000_LANG') in ('UKR') then ist.ist_name_num_ukr_f else ist.ist_name_num_rus_f end as ist_num_f,
       dis_pltec.pltec_common.GetItmName2Date(p_itm => doci.doci_itm) as doci_itm_list
from document doc
inner join v_ddn_doc_st dst on dst.dic_value = doc.doc_st
inner join v_doc_type_ur dct on dct.dct_id = doc.doc_dct
inner join doc_spc dspc on dspc.dspc_doc = doc.doc_id and dspc.dspc_rel = 'EXEC'
inner join doc_item doci on doci.doci_doc = doc.doc_id and doci.doci_kind in ('ITM')
inner join vm_item_storage ist on ist.ist_id = doci.doci_ist_place
inner join item itm on doci.doci_itm = itm.itm_id
where doc.doc_rec_st in 'A'
      and doc.doc_dpr = v('P1000_FIRM')
/*filter_condition_begin*/
#=filter_polygon_grp#
#=filter_polygon_type#
#=filter_polygon_num#
#=filter_polygon_st#
#=filter_polygon_dt#
#=filter_polygon_ist#
#=filter_polygon_itmg#
#=filter_polygon_itm#
#=filter_polygon_spc#
/*filter_condition_end*/

// js file
require('/i/pltec/common/doc_polygon.css', plasmisui_version);

// Функція відкриття картки для інших сторінок
function onClickPolygon(doc_id) {
  if (typeof Rest_Itm == 'undefined') {
    require('/i/pltec/common/doc_polygon.js').then((e) => {
      const addnewAddDocPolygonWndw = new AddDocPolygonWndw({ mode: "update", grid: null });
      addnewAddDocPolygonWndw._docId = doc_id;
      addnewAddDocPolygonWndw.open({ mode: 'update', grid: null });
    });
  } else {
    const addnewAddDocPolygonWndw = new AddDocPolygonWndw({ mode: "update", grid: null });
    addnewAddDocPolygonWndw._docId = doc_id;
    addnewAddDocPolygonWndw.open({ mode: 'update', grid: null });
  }
}

// Головний grid
function getPlgGrid() {
  let f_toolbar = [];
  let f_srt = [];
  let f_clmn = [];
  let f_grp = [];
  f_toolbar.push({
    name: "add_plg",
    template: "<button type='button' id='add_plg' class='k-button' title='" + lang("Додати") + "'><img src='/i/pltec/common/fatcow-hosting-icons-3000/16x16/add.png' alt='" + lang("Додати") + "'></button>"
  });
  f_toolbar.push({
    name: `context_menu_but`,
    template: "<button type='button' id='context_menu_but' class='k-button' title=' ${lang("Опєпаиї")} ' name='context_menu_but'><img src='/i/pltec/common/operation16p.png' alt=' ${lang("Опєпаиї")} ' ></button>"
  });
  f_toolbar.push({
    name: "search"
  });
  f_grp.push({
    field: "ist_par",
    dir: "asc"
  });
};
```



```

f_srt.push({
  field: "ist_num_f",
  dir: "asc"
})
f_clmn.push({
  field: "doc_st_sname",
  title: lang("Статус"),
  template: '',
  headerTemplate: "",
  headerAttributes: {
    title: lang("Статус")
  },
});
f_clmn.push({
  field: "doc_id",
  title: lang("Ід"),
  headerTemplate: lang("Ід"),
  headerAttributes: {
    title: lang("Ід")
  },
});
f_clmn.push({
  field: "doc_num",
  title: lang("Номер"),
  headerTemplate: lang("Номер"),
  headerAttributes: {
    title: lang("Номер")
  },
});
f_clmn.push({
  field: "dct_name",
  title: lang("Тип документу"),
  headerTemplate: lang("Тип документу"),
  headerAttributes: {
    title: lang("Тип документу")
  },
});
f_clmn.push({
  field: "doc_dt",
  title: lang("Дата створення документу"),
  headerTemplate: lang("Дата"),
  headerAttributes: {
    title: lang("Дата створення документу")
  },
  format: "{0: dd.MM.yy}",
  footerAttributes: {
    style: "text-align:center;"
  },
});
f_clmn.push({
  field: "dpr_name_full",
  title: lang("Підприємство"),
  headerTemplate: lang("Підприємство"),
  headerAttributes: {
    title: lang("Підприємство")
  },
});
f_clmn.push({
  field: "ist_place",
  title: lang("Місце зберігання") + ': ' + lang("повно"),
  headerTemplate: lang("Місце зберігання") + ': ' + lang("повно"),
  headerAttributes: {
    title: lang("Місце зберігання") + ': ' + lang("повно")
  },
});
f_clmn.push({
  field: "ist_par",
  title: lang("Місце зберігання") + ': ' + lang("пращур"),
  headerTemplate: lang("Місце зберігання") + ': ' + lang("пращур"),
  headerAttributes: {
    title: lang("Місце зберігання") + ': ' + lang("пращур")
  },
});
f_clmn.push({
  field: "ist_num_f",
  title: lang("Місце зберігання"),
  headerTemplate: lang("Місце зберігання"),
  headerAttributes: {
    title: lang("Місце зберігання")
  },
});
f_clmn.push({
  field: "doci_itm_list",
  title: lang("МЦ"),
  headerTemplate: lang("МЦ"),
  headerAttributes: {
    title: lang("МЦ")
  },
},

```

```

width: 150,
hidden: false,
attributes: {
  title: "#=doci_itm_list#",
  style: "text-align:left; white-space:nowrap;"
});
f_clmn.push({
  field: "doc_spc",
  title: lang("Виконавець"),
  headerTemplate: lang("Виконавець"),
  headerAttributes: {
    title: lang("Виконавець")
  },
});
f_clmn.push({
  field: "doc_name",
  title: lang("Примітки"),
  headerTemplate: lang("Примітки"),
  headerAttributes: {
    title: lang("Примітки")
  },
});
f_clmn.push({
  field: "edit",
  title: lang("Редагування"),
  headerTemplate: "<a title='" + lang("Редагування") + "'></a>",
  headerAttributes: {
    title: lang("Редагування")
  },
  width: 40,
  hidden: false,
  template: '',
  attributes: {
    style: "cursor: pointer; text-align:center"
  }
});
f_clmn.push({
  field: "delete",
  title: lang("Видалення"),
  headerTemplate: "<a title='" + lang("Видалення") + "'></a>",
  headerAttributes: {
    title: lang("Видалення")
  },
  width: 40,
  hidden: true,
  template: '',
  attributes: {
    style: "cursor: pointer; text-align:center"
  }
});
//Операції
function Operations(grid) {
  window.operations_menu = $("<ul id='Operations_Menu_id' </ul>").kendoContextMenu({
    target: $(grid.element).find("#context_menu_but"),
    showOn: "click",
    open: e => { onOpen(e, grid) },
    select: e => { onSelect(e, grid) },
    dataSource: [{
      text: "<label class='dlink'>" + lang("Видалити") + "</label>",
      encoded: false,
      imageUrl: "/i/pltec/common/delete16.gif"
    },]
  }).data('kendoContextMenu');
  $('ul[data-role="contextmenu"] span').attr('style', 'white-space: nowrap;');
};
var onOpen = function (e, grid) {
  if (!grid.dataItem(grid.select())) {
    $('ul[data-role="contextmenu"] li').each(function (i) {
      if ($(this).find("label")[0].className == 'dlink') {
        $(this).addClass('k-state-disabled')
      }
    });
  } else $('ul[data-role="contextmenu"] li').each(function (i) {
    if ($(this).find("label")[0].className == 'dlink') {
      $(this).removeClass('k-state-disabled')
    }
  });
};
function onSelect(e, grid) {
  var SelectLbl = $(e.item).find("label")[0].className;
  if (SelectLbl) SelectedOper(SelectLbl, e, grid);
}

```

```

var SelectedOper = function (buttonClass, e, grid) { //Ф-ції на кнопки
    var selectedItem = grid.dataItem(grid.select());
    if (buttonClass == 'dlink') { //Видалити
        console.log('dlink')
        if (confirm(lang("Ви впевнені, що бажаєте видалити?"))) {
            let values = [];
            values.push({
                "doc_id": selectedItem.doc_id,
                "doc_obi_ts": selectedItem.doc_obi_ts,
            });
            apex.server.process("DocPolygon_WEB", {
                x01: JSON.stringify(values),
                x02: 'delete'
            }, {
                dataType: "json",
                success: (pData) => {
                    if (pData.err == 0) {
                        noty({ text: 'Дані видалено', layout: 'topRight', type: 'SUCCESS', timeout: 2500 });
                        grid.dataSource.read();
                    } else {
                        noty({ text: pData.err_mess, layout: 'topRight', type: 'WARNING', timeout: 2500 });
                    }
                }
            });
        }
    }
}

// Вікно додавання полігону
class AddDocPolygonWndw {
    constructor(mode, grid) {
        const that = this;
        this.grid = grid;
        this.mode = mode;
        this.refreshGrid = true;
        this._windowId = `AddDocPolygonWndw`;
        this._docId;
        this._docObiTs;
        this.doc_id = "";
        this.bc_id = "";
        this._docSt;
        this._dctCode;
        this._dctName;
        console.log('start AddDocPolygonWndw');
        // шаблон для обов'язкових полів
        that.main_field =
            `<div class='tooltip'>
                <label style='color:red; cursor:pointer'>*</label>
                <span class='tooltiptext' style='margin-top:0px; margin-left:-35px;'>` + lang("Обов'язкове поле") + `</span>
            </div>`;
        // html головних елементів вікна
        that.mainContainerWindow = `(<div id='add_doc_plg_wndw' style='display: none;'></div>");
        that.interContainer = `(<div class='plg-main_' id='plg_container_'></div>");
        that.headContainer = `(<div id='head_doc_plg_container'></div>");
        that.footerContainer = `(<div id='footer_doc_plg_container'></div>");
        that.mainToolbarWindow = `(<div class='box-plg-1 box-polygon'></div>");
        // вставка головних елементів вікна на сторінку
        that.mainContainerWindow.append(that.interContainer); // вікно
        that.headContainer.append(that.mainToolbarWindow);
        that.interContainer.append(that.headContainer);
        $("body").append(that.mainContainerWindow); // тулбар вікна
        // накидаєм елементи тулбару та формуєм його на вікно
        that.tlbr = [{
            name: "edit_doc", id: 'edit_doc',
            template: "<button style='margin-left: 2px;' class='edit_doc k-button' type='button' name='edit_doc' title='\" + lang('Редагувати') + \"'><img src='/i/pltec/common/doc_st_16_edit.gif'></button>"
        }, {
            name: "save_doc", id: 'save_doc',
            template: "<button class='save_doc k-button' type='button' name='save_doc' title='\" + lang('Зберегти') + \"'><img src='/i/pltec/common/save16.gif'></button>"
        }, {
            name: "cancel_doc", id: 'cancel_doc',
            template: "<button class='cancel_doc k-button' type='button' name='cancel_doc' title='\" + lang('Відмінити') + \"'><img src='/i/pltec/common/fatcow-hosting-icons-3.9.2/FatCow_Icons16x16/cancel.png'></button>"
        }, {
            name: "header_oper_doc", id: 'header_oper_doc',
            template: "<button class='header_oper_doc k-button' type='button' name='header_oper_doc' title='\" + lang('Операції') + \"'><img src='/i/pltec/common/operation16p.png'></button>"
        }, {
            name: "doc_print", id: 'doc_print',
            template: "<button class = 'doc_print k-button' type='button' name='doc_print'><img src='/i/pltec/common/printer.png' style='margin-right: 5px;' title='\" + \"' + lang('Друк бірки') + \"'>\" + \"' + lang('Друк бірки') + \"</button>"
        }
    ]
}

```

```

}, {
  name: "headers_show_doc_plg", id: 'headers_show_doc_plg',
  template: "<button style='margin-right: 1px;' class='headers_show_doc_plg k-button' type='button' name='headers_show_doc_plg' title='" +
    lang('Показати') + " / '" + lang('сховати') + "' " + lang('інформацію картки') +
    "'><img src='/i/pltec/common/all_page_mode.png'></button>" +
    "<button class='card_instruction_plg k-button' type='button' name='card_instruction_plg' title='" + lang('Інструкція')
    + "'><img src='dis_web/r/files/static/v1/Y/manual.gif'></button>"
  });
that.mainToolBarWindow.kendoToolBar({
  items: that.tlbr
});
that.head_elements = `<div class='plg_head_container'>
  <div class='group-plg--container'>
    <div class='plg_class_line plg_class_line_1'>
      <div class='box_for_plg'>
        <label> ${lang('Home')} ${that.main_field} </label>
        <div>
          <input class='k-textbox' id='doc_num_plg' type='text' style="width: 99%;">
        </div>
      </div>
      <div class='box_for_plg'>
        <label class = 'data-picker'> ${lang('Дата')} ${that.main_field} </label>
        <div>
          <input class = 'k-sel' data-role='datepicker' id='doc_dt_plg' type='text' style="width: 99%;">
        </div>
      </div>
      <div class='box_for_plg'>
        <label> ${lang('Тім')} ${that.main_field} </label>
        <div style="flex-grow: 1;">
          <select class = 'k-sel' id='doc_type_plg' type='text' style="width: 99%;"></select>
        </div>
      </div>
      <div class='box_for_plg'>
        <label> ${lang('Статус')} ${that.main_field} </label>
        <div>
          <select class = 'k-sel' id='doc_st_plg' type='text' style="width: 99%;"></select>
        </div>
      </div>
    </div>
    <div class='plg_class_line plg_class_line_2'>
      <div class='box_for_plg'>
        <label> ${lang('Підприємство')} ${that.main_field} </label>
        <select class = 'k-sel' id='doc_dpr_plg' type='text'></select>
      </div>
      <div class='box_for_plg'>
        <label> ${lang('Місце зберігання')} ${that.main_field} </label>
        <select class = 'k-sel'; id='doc_ist_place' type='text'></select>
        <button type='button' id='wndw-class-btn_add_place' style='height:25.5px;' class='k-button' title=' ' + lang('Додати місце
зберігання') + ' '><img src='/i/pltec/common/fatcow-hosting-icons-3000/16x16/add_jur.png'></button>
      </div>
    </div>
    <div class='plg_class_line plg_class_line_6'>
      <div class='box_for_plg'>
        <label> ${lang('МЦ')} </label>
        <select class = 'k-sel'; id='itm_name_plg' type='text' ></select>
      </div>
    </div>
    <div class='plg_class_line plg_class_line_4'>
      <div class='box_for_plg'>
        <label> ${lang('Виконавець')} ${that.main_field} </label>
        <select class = 'k-sel' id='doc_spc_plg' type='text'></select>
      </div>
    </div>
    <div class='plg_class_line plg_class_line_5'>
      <div class='box_for_plg'>
        <label> ${lang('Примітки')} </label>
        <input class = 'k-textbox'; id='doc_comment_plg' style = 'resize: none; height: 24px;' ></input>
        <button type='button' id='wndw-class-btn_comment_edit' style='height:25.5px;' class='k-button' title=' '><img
src='/i/pltec/common/notepad_16.gif'></button>
      </div>
    </div>
    <div class='instruction_class_line_6'>
      <div class='box_for_instruction'>
        <div id='plg_tabstrip_place'></div>
      </div>
    </div>
  </div>
  <div class='instruction_class_line_6' id='plg_tabstrip_place'></div>";
that.doc_head = $(that.head_elements);
that.interContainer.append(that.doc_head, that.footerContainer);
that.tabStrip = $("<div class='instruction_class_line_6' id='plg_tabstrip_place'></div>");
that.footerContainer.append(that.tabStrip);

```



```

        $('#obj2dcc').data('kendoGrid').dataSource.read();
    }
    //сповіщення про знайдені елементи
    apex.server.process("SendMess4ProdPlgn", {
        x01: doc_id_prd
    }, {
        dataType: "text",
        success: (pData) => {
        }
    });
    });
    } else {
        //додавання файлу до протокола та вилучення показників
        apex.server.process("ExportDmsp4Expert", {
            x01: doc_id_prd,
            x02: result.odc_id,
            x03: dmspk_id,
            x04: 'ALL'
        }, {
            dataType: "text",
            success: (pData) => {
                kendo.ui.progress(that.dialog.element, false);
                grid_dmsp.dataSource.read();
                if ($('#obj2dcc').length) {
                    $('#obj2dcc').data('kendoGrid').dataSource.read();
                }
                //сповіщення про знайдені елементи
                apex.server.process("SendMess4ProdPlgn", {
                    x01: doc_id_prd
                }, {
                    dataType: "text",
                    success: (pData) => {
                    }
                });
            }
        });
    }
} else if (SelectLbl === 'delete_polygon') {
    if (confirm(lang("Ви впевнені, що бажаєте видалити?"))) {
        let values = [];
        values.push({
            "doc_id": selectedItem.doc_id,
            "doc_obi_ts": selectedItem.doc_obi_ts,
        });
        apex.server.process("DocPolygon_WEB", {
            x01: JSON.stringify(values),
            x02: 'delete'
        }, {
            dataType: "json",
            success: (pData) => {
                if (pData.err == 0) {
                    noty({ text: 'Дані видалено', layout: 'topRight', type: 'SUCCESS', timeout: 2500 });
                    that.is_update = true;
                    that.close();
                    that.grid.dataSource.read();
                } else {
                    noty({ text: pData.err_mess, layout: 'topRight', type: 'WARNING', timeout: 2500 });
                }
            }
        });
    }
}
dataSource: [
    {
        text: '<label style=cursor:pointer; class=add_protocol_for_polygon> ${lang("Створити протокол та завантажити показники")}'
    }
],
    imageUrl: "/i/pltec/common/fatcow-hosting-icons-3.9.2/FatCow_Icons16x16/book_open.png",
    encoded: false
}, {
    text: '<label style=cursor:pointer; class=delete_polygon> ${lang("Видалити")}'
}, {
    imageUrl: "/i/pltec/common/delete16.gif",
    encoded: false
}
]);
}).data('kendoContextMenu');
that.plgHeadersShowCtxtMenu = $("<ul id=plg_headers_show_ctxt_menu' >").kendoContextMenu({
    target: $(button.headers_show_doc_plg),
    showOn: "click", // mouseenter
    alignToAnchor: true,
    animation: {
        open: {
            effects: "none"
        },
    },
    activate: function (e) {
        let newLeftRetreat = e.sender.target[0].getBoundingClientRect().left - e.sender.element.parent().width() + e.sender.target[0].offsetWidth
        e.sender.element.parent().css('left', newLeftRetreat);
    }
});

```

```

    },
    select: function (e) {
        let checkLbl = $(e.item).find("label")[0].className;
        that.SelectPlgHeadersShowCtxMenu(checkLbl)
        localStorage.setItem('plg_headers_show_ctxt_menu', checkLbl);
    },
    dataSource: [{
        text: "<label style='cursor:pointer;' class='plg_header_region_all'>" + lang("Показати все") + "</label>",
        encoded: false,
        imageUrl: "/i/pltec/common/all_page_mode.png"
    }, {
        text: "<label style='cursor:pointer;' class='plg_header_region_midle'>" + lang("Головну інформацію та закладки") + "</label>",
        encoded: false,
        imageUrl: "/i/pltec/common/all_page_mode_mid.png"
    }, {
        text: "<label style='cursor:pointer;' class='plg_header_region_down'>" + lang("Тільки закладки") + "</label>",
        encoded: false,
        imageUrl: "/i/pltec/common/all_page_mode_tabs.png"
    }
    ]
}).data("kendoContextMenu");
this.dropDownDctType = $(this.doc_head).find("#doc_type_plg").kendoDropDownList({
    dataTextField: "name",
    dataValueField: "value",
    optionLabel: " ",
    suggest: true,
    autoBind: true,
    autoWidth: true,
    dataSource: {
        transport: {
            read: function (options) {
                apex.server.process("GetDocMailTypeFilter", {
                    x01: 'DVL_BURIAL'
                }, {
                    dataType: "json",
                    success: function (pData) {
                        options.success(pData);
                    }
                });
            }
        }
    }
}).data("kendoDropDownList");

/* Дата */
this.dropDatePickerDocDt = $(this.doc_head).find("#doc_dt_plg").kendoDatePicker({ format: 'dd.MM.yyyy' }).mask("99.99.9999");
this.multiSelectDocSpc = $(this.doc_head).find("#doc_spc_plg").kendoMultiSelect({
    dataTextField: "label_s",
    dataValueField: "id",
    filter: "contains",
    autoBind: false,
    maxSelectedItems: 1,
    minLength: 3,
    dataSource: {
        serverFiltering: true,
        transport: {
            read: function (options) {
                apex.server.process("AutocompleteSpc",
                    {
                        x01: function () { return that.doc_head.find("#doc_spc_plg").data("kendoMultiSelect").input.val() },
                        x02: function () { return that._doc_spc_plg ? that._doc_spc_plg : " " },
                    }, {
                        dataType: "json",
                        success: function (pData) {
                            options.success(pData);
                        }
                    }
                );
            }
        },
        change: function (e) {
        },
    }
}).data("kendoMultiSelect");

//Cratyc
this.dropDownDocSt = $(this.doc_head).find("#doc_st_plg").kendoDropDownList({
    dataTextField: "d",
    dataValueField: "r",
    minLength: 1,
    suggest: true,
    autoBind: true,
    autoWidth: true,
    optionLabel: " ",
    dataSource: {
        transport: {
            read: function (options) {
                apex.server.process("GetSt4TechTask", {}, {
                    dataType: "json",

```

```

        success: function (pData) {
            options.success(pData);
        });},},},
    }).data('kendoDropDownList');
    this.multiSelectItm = $(this.doc_head).find("#itm_name_plg").kendoDropDownList({
        dataTextField: "itm_name",
        dataValueField: "itm_id",
        dataSource: {
            transport: {
                read: function (options) {
                    apex.server.process("GetItm4Polygon", {}, {
                        dataType: "text",
                        success: function (pData) {
                            var json = jQuery.parseJSON(pData);
                            options.success(json);
                        }
                    });},},},
                group: { field: "itm_name" }
            },
            height: 400,
            open: function (e) {
                var listContainer = e.sender.list.closest(".k-list-container");
                listContainer.width(listContainer.width() + kendo.support.scrollbar());
            },
            change: function (e) {
            }
        }).data('kendoDropDownList');
    this.uiDocIstPlaceTree = $(this.doc_head).find("#doc_ist_place").kendoDropDownTree({
        dataTextField: "name",
        dataValueField: "value",
        dataSource: {
            transport: {
                read: function (options) {
                    apex.server.process('GetIst4PolygonWndw', {}, {
                        dataType: 'json',
                        success: function (data) {
                            options.success(data);
                        }
                    });},},},
            autoBind: false,
            autoClose: false,
            autoWidth: true,
            height: 500,
            filter: "contains",
            checkboxes: {
                template: function (data) {
                    // додаємо чекбокси, сховуючи відображення по умові з hide_checkbox
                    let kendoOutput = '<input type=\'checkbox\' tabindex=\'-1\' ' +
                        ((data.item.enabled === false) ? 'disabled\' : ') + ' ' +
                        (data.item.checked ? 'checked\' : ') + ' id=\'_' + (data.item.uid) + ' \' class=\'k-checkbox\' ' +
                        (data.item.hide_checkbox ? 'style=\'display: none; margin: -7px 0px 0px 0px;\' ' : 'style=\'margin: -7px 0px 0px 0px;\' ') +
                        ' /><span class=\'k-checkbox-label checkbox-span\'></span>';
                    return kendoOutput;
                },
                open: function (e) {
                    // перше відкриття віджету
                    if (!e.sender.isFirstOpen) {
                        console.log('first open');
                        // додаємо задній текст на інпут фільтрації
                        e.sender.filterInput.attr("placeholder", lang("Кінцевий підрозділ містить") + ":");
                        // розкриваємо всі дочірні елементи
                        e.sender.tree.children().find(".k-icon.k-i-expand:first").trigger('click');
                        // трігерим флаг першого відкриття
                        e.sender.isFirstOpen = true;
                        // додаємо значок звернути/розгорнути всі дочірні елементи
                        let expandButton = `<span style="margin-left: -45px;"
                            class="k-link k-link-refresh" aria-label="Refresh the Search" title="${lang('згорнути')} + '/' + lang('розгорнути')} + ' ' +
                            lang('всі дочірні елементи')">
                            <span unselectable="on" class="k-klevis k-icon k-i-expand" style="font: 20px/1 WebComponentsIcons; margin-top: -
                            2px;"></span></span>`;
                        e.sender.filterInput.after(expandButton)
                        // додаємо подію згортання (якщо знайшли хоча б 1 розгорнутий елемент не враховуючи перший - загальний) / розгортання всіх
                        дочірніх елементів
                        e.sender.filterInput.siblings('span').children('.k-klevis.k-icon.k-i-expand').off('click').on('click', function (e) {
                            if ($(e.target).closest('.k-list-filter').siblings('.k-widget.k-treeview').find('.k-icon.k-i-collapse:not(:first)').length) {
                                $(e.target).closest('.k-list-filter').siblings('.k-widget.k-treeview').find('.k-icon.k-i-collapse:not(:first)').trigger('click');
                            } else {
                                $(e.target).closest('.k-list-filter').siblings('.k-widget.k-treeview').find('.k-icon.k-i-expand').trigger('click');
                            }
                        });
                        // наступні відкриття віджету
                    } else {

```



```

        console.log('second open');
    }},
    select: function (e) {
        // забираємо тригер для полів де прихований чекбокс
        if ($(e.node).children('div').find('input').css('display') == 'none') {
            e.preventDefault();
        },
    },
    change: function (e) {
        // якщо обрано більше 1 поля - обираємо останнє та тригерим change
        if (this.value().length > 1) {
            let last_item = this.value().slice(-1)
            e.sender.value([]);
            e.sender.value(last_item)
            e.sender.trigger('change')
        } else {
            // для всіх попередньо виділених жовтим ітемів - забираємо окрас через видалення класу
            e.sender.tree.children().find('span.k-in.k-state-selected').removeClass('k-state-selected')
            // шукаємо щойно чекнутий чекбокс та красим текст біля нього в жовтий (виділення)
            e.sender.tree.children().find('input[type="checkbox"]:checked').closest('span.k-checkbox-wrapper').siblings('span.k-in').addClass('k-state-selected')
        }
    }
}).data('kendoDropDownTree');

this.dropDownDprPlg = $(that.doc_head).find("#doc_dpr_plg").kendoDropDownList({
    optionLabel: lang("Виберіть підприємство") + "...",
    dataTextField: "dpr_name",
    dataValueField: "dpr_id",
    filter: "startswith",
    value: $v("P0_FIRM"),
    dataSource: {
        transport: {
            read: function (options) {
                apex.server.process("GetFirm",
                {
                }, {
                    dataType: "json",
                    success: function (pData) {
                        options.success(pData);
                    }
                });
            }
        }
    }
}).data("kendoDropDownList");

this.doc_head.find('#wndw-class-btn_add_place').off().on('click', () => {
    const addnewIstPlaceAddWindow = new IstPlaceAddWindow({ mode: "insert", grid: grid });
    addnewIstPlaceAddWindow.backHandler = () => {
        that.uiDocIstPlaceTree.dataSource.read();
    };
    addnewIstPlaceAddWindow.open({ mode: 'insert', grid: grid })
});

// Вікно Обрання типу
class PolygonTpWindow {
    constructor(mode, grid) {
        const that = this;
        this.grid = grid;
        this.mode = mode;
        this.refreshGrid = true;
        this._windowId = `PolygonTpWindow`;
        this._dctCode;
        console.log('start PolygonTpWindow');
        this.PolygonTpWindow = $('<div id = "${this._windowId}" style="display: none;"></div>');
        this.interContainer = $('<div class = "TechTpMain" id="container_${this._windowId}"></div>');
        this.PolygonTpWindow.append(this.interContainer);
        $("body").append(this.PolygonTpWindow);
        /*Тип*/
        this.polygon_tp = $('<div class="box-polygon-1 box-polygon"><label for = 'polygon_tp' style = 'margin-right: 5px;'> + lang("Тип") +
            "<div class='tooltip'><label style='color: red; cursor: pointer'>&nbsp;*</label><span class='tooltiptext' style = 'margin-top: 20px;'>\
            " + lang("Обов'язкове поле") + "</span></div></label><input class = 'k-sel'; style = 'width: 92%;' id='dropDownPolygonTp' type='text'
        </div>");
        this.polygonBtnFooter = $('<br><div id = 'polygon_footer' class = 'boxBtnFooter style="margin-top: 5px;'>\
            <div class = 'boxBtnFooter' style = 'float: right'>\
                <button class = 'polygon_footer' id='polygon_add' style = 'float: left' > + lang("Створити") + "</button>\
                <button class = 'polygon_footer' id='polygon_close' style = 'float: right'> + lang("Закрити") + "</button>\
            </div>\
        </div>");
        this.interContainer.append(this.polygon_tp, this.polygonBtnFooter);
        this.dropDownPolygonTp = $(this.polygon_tp).find("#dropDownPolygonTp").kendoDropDownList({
            dataTextField: "name",
            dataValueField: "value",

```

```

suggest: true,
autoBind: true,
autoWidth: true,
dataSource: {
  transport: {
    read: function (options) {
      apex.server.process("GetDocMailTypeFilter", {
        x01: 'DVL_BURIAL'
      }, {
        dataType: "json",
        success: function (pData) {
          options.success(pData);
        }
      });
    }
  }
}).data("kendoDropDownList");
this.polygonCntAdd = $(this.polygonBtnFooter).find("#polygon_add").kendoButton({
  imageUrl: "/i/pltec/common/fatcow-hosting-icons-3.9.2/FatCow_Icons16x16/accept_button.png",
  click: function (e) {
    const addnewAddDocPolygonWndw = new AddDocPolygonWndw({ mode: "insert", grid: that.grid });
    addnewAddDocPolygonWndw._dctCode = that.dropDownPolygonTp.value();
    addnewAddDocPolygonWndw._dctName = that.dropDownPolygonTp.text();
    addnewAddDocPolygonWndw.open({ mode: 'insert', grid: that.grid });
    that.close();
  }
}).data("kendoButton");
this.techTaskCntClose = $(this.polygonBtnFooter).find("#polygon_close").kendoButton({
  imageUrl: "/i/pltec/common/fatcow-hosting-icons-3.9.2/FatCow_Icons16x16/cancel.png",
  click: function (e) {
    that.close();
  }
}).data("kendoButton");
set dctCode(value) {
  this._dctCode = value;
}
get dctCode() {
  return this._dctCode;
}
open(options) {
  let that = this;
  this.dialog.center();
  this.dialog.open();
  this.grid = options.grid;
  this.mode = options.mode;
  this.dialog.title(lang("Полігони") + ': ' + lang("Додавання"));
}
close() {
  this.dialog.close();
}
}
that.tabstrip = $("
```

```

});
that.tabstrip.select(0);
};

//grid Показників
function GetDmsp4Polygon(doc_id) {
    f_toolbar = [];
    f_clmn = [];
    f_grp = [];
    f_srt = [];
    let $gridDmsp4Polygon = $("<div id = 'grid_dmsp_dpt'></div>").kendoGrid({
        dataSource: {
            transport: {
                read: function (options) {
                    apex.server.process("GetDmsp4Poligon", {
                        x01: doc_id
                    }, {
                        success: function (pData) {
                            options.success(pData);
                        }
                    });
                },
            },
            schema: {
                model: {
                    id: "doc_id",
                    serverPaging: false,
                    serverSorting: false,
                    serverFiltering: false,
                    sort: [
                        { field: "dmsp_order", dir: "asc" },
                        { field: "dmsp_num", dir: "asc" },
                        { field: "dmsp_group", dir: "asc" }
                    ],
                    group: [
                        { field: "doc_num", dir: "asc" },
                        { field: "plt_name", dir: "asc" },
                    ]
                },
            },
            toolbar: f_toolbar,
            columnMenu: true,
            columns: [
                {
                    field: "dmsp_id",
                    title: lang("Ід"),
                    headerTemplate: lang("Ід"),
                    headerAttributes: {
                        title: lang("Ід")
                    },
                },
                {
                    field: "plt_name",
                    title: lang("Місце"),
                    headerTemplate: lang("Місце"),
                    headerAttributes: {
                        title: lang("Місце")
                    },
                },
                {
                    field: "dmsp_order",
                    title: "Сортування",
                    headerTemplate: lang("Сортування"),
                    headerAttributes: {
                        title: lang("Сортування")
                    },
                },
                {
                    field: "dmsp_group",
                    title: "Група",
                    headerTemplate: lang("Група"),
                    headerAttributes: {
                        title: lang("Група")
                    },
                },
                {
                    field: "doc_num",
                    title: "№",
                    headerTemplate: "№",
                    headerAttributes: {
                        title: "№"
                    },
                },
            ],
            width: 120,
            template: '#=doc_num#',

```

```

hidden: true,
attributes: {
  title: "#=doc_num#",
  style: "text-align:left; white-space:nowrap;"
}
},
{
  field: "dmsp_name",
  title: "Назва",
  headerTemplate: lang("Назва"),
  headerAttributes: {
    title: lang("Назва")
  },
  width: 120,
  template: "#=dmsp_name#",
  attributes: {
    title: "#=dmsp_name#",
    style: "text-align:left; white-space:nowrap;"
  }
},
{
  field: "itprc_name",
  title: lang("Показник"),
  hidden: true,
  headerTemplate: lang("Показник"),
  headerAttributes: {
    title: lang("Показник")
  },
  width: 170,
  template: "#=itprc_name#",
  attributes: {
    title: "#=itprc_name#",
    style: "text-align:left; white-space:nowrap;"
  }
},
{
  field: "itprc_code",
  title: "Код",
  hidden: true,
  headerTemplate: lang("Код"),
  headerAttributes: {
    title: lang("Код")
  },
  width: 100,
  template: "#=itprc_code#",
  attributes: {
    title: "#=itprc_code#",
    style: "text-align:left; white-space:nowrap;"
  }
},
{
  field: "unt_name",
  title: "Одиниці вимірювання",
  headerTemplate: lang("Одиниці вимірювання"),
  headerAttributes: {
    title: lang("Одиниці вимірювання")
  },
  width: 100,
  template: "#=unt_name#",
  attributes: {
    title: "#=unt_name#",
    style: "text-align:left; white-space:nowrap;"
  }
},
{
  field: "dic_sname_tp",
  title: "Тип параметру",
  headerTemplate: lang("Тип параметру"),
  headerAttributes: {
    title: lang("Тип параметру")
  },
  width: 70,
  template: "#=dic_sname_tp#",
  attributes: {
    title: "#=dic_sname_tp#",
    style: "text-align:left; white-space:nowrap;"
  }
},
{

```

```

        field: "check_tp",
        title: "Тип перевірки",
        headerTemplate: lang("Тип перевірки"),
        headerAttributes: {
            title: lang("Тип перевірки")
        },
        width: 70,
        template: '#=check_tp#',
        attributes: {
            title: "#=check_tp#",
            style: "text-align:left; white-space:nowrap;"
        }
    },
    {
        field: "dmsp_min",
        title: "Мінімальне значення",
        headerTemplate: lang("Мінімальне значення"),
        headerAttributes: {
            title: lang("Мінімальне значення")
        },
        width: 65,
        template: '#=nvl(dmsp_min,`')#',
        attributes: {
            title: "#=nvl(dmsp_min,`')#",
            style: "text-align:right; white-space:nowrap;"
        }
    },
    {
        field: "dmsp_max",
        title: "Максимальне значення",
        headerTemplate: lang("Максимальне значення"),
        headerAttributes: {
            title: lang("Максимальне значення")
        },
        width: 65,
        template: '#=nvl(dmsp_max,`')#',
        attributes: {
            title: "#=nvl(dmsp_max,`')#",
            style: "text-align:right; white-space:nowrap;"
        }
    },
    {
        field: "dmsp_value",
        title: lang("Значення"),
        headerTemplate: lang("Значення"),
        headerAttributes: {
            title: lang("Значення")
        },
        width: 80,
        template: '<input id="val_dmsp_#=dmsp_id#" class="val_dmsp k-textbox" value="#=nvl(dmsp_value,`')#" style="width:100%; margin-left: 0px; height: 25px; text-align:right; disabled="disabled">',
        attributes: {
            style: "text-align:right; white-space:nowrap;"
        }
    },
    {
        field: "dmsp_descript",
        title: lang("Опис"),
        headerTemplate: lang("Опис"),
        headerAttributes: {
            title: lang("Опис")
        },
        width: 110,
        hidden: true,
        attributes: {
            style: "text-align:right; white-space:nowrap;"
        }
    }, {
        field: "val_calc",
        title: lang("Розрахунок") + " , %",
        headerTemplate: lang("Розрахунок") + " , %",
        headerAttributes: {
            title: lang("Розрахунок") + " , %"
        }
    },
    },
    ],
    dataBound: function (e) {
        let grid = e.sender,
            grid_el = grid.element;
        $(grid_el).find($('.val_dmsp')).each(function () {

```

```

let dataItem = grid.dataItem($(this).closest("tr"));
if (dataItem.style == 'valid') {
    style_input = 'width:100%; margin-left: 0px; height: 25px;';
    $(this).attr('style', style_input);
    if (dataItem.dmsp_check_tp != "DESCRIPT" && dataItem.dmsp_check_tp != "LIST" && dataItem.dmsp_check_tp != "DATE") {
        style_input = 'border-color: green; width:100%; margin-left: 0px; height: 25px; text-align:right;';
        $(this).attr('style', style_input);
    };
} else if (dataItem.style == 'error') {
    style_input = 'background-color:#93f993 !important; width:100%; margin-left: 0px; height: 25px;';
    if (dataItem.need_calc == 'T') {
        $(this).closest("tr").find("#val_calc_" + dataItem.dmsp_id).closest("td").attr('style', style_input);
    } else {
        $(this).attr('style', style_input);
    }
    if (dataItem.dmsp_check_tp != "DESCRIPT" && dataItem.dmsp_check_tp != "LIST" && dataItem.dmsp_check_tp != "DATE" &&
dataItem.need_calc != "T") {
        style_input = 'background-color:#93f993 !important; border-color: green; width:100%; margin-left: 0px; height: 25px; text-
align:right;';
        $(this).attr('style', style_input);
    };
} else if (dataItem.style == 'exep') {
    style_input = 'background-color:#93f993 !important; width:100%; margin-left: 0px; height: 25px;';
    $(this).attr('style', style_input);
    if (dataItem.dmsp_check_tp != "DESCRIPT" && dataItem.dmsp_check_tp != "LIST" && dataItem.dmsp_check_tp != "DATE") {
        style_input = 'background-color:#93f993 !important; border-color: green; width:100%; margin-left: 0px; height: 25px; text-
align:right;';
        $(this).attr('style', style_input);
    };
};
};
if (dataItem.style == " || dataItem.style == 'noerror') {
    if (dataItem.dmsp_check_tp != "DESCRIPT" && dataItem.dmsp_check_tp != "LIST" && dataItem.dmsp_check_tp != "DATE") {
        style_input = 'border-color: green; width:100%; margin-left: 0px; height: 25px; text-align:right;';
        $(this).attr('style', style_input);
    };
};
});
contextMenuDptPolygon(grid);
removeSpinner();
});
var grid = $gridDmsp4Polygon.data("kendoGrid");
return grid;
}

//побудова меню операцій
function contextMenuDptPolygon(grid) {
    var f_lst = [];
    let grid_el = grid.element;
    f_lst.push({
        text: '<label style='\"cursor:pointer;\" class='\"updt_bc_st'> ${lang("Змінити статус")} </label>',
        imageUrl: "/i/pltec/common/fatcow-hosting-icons-3.9.2/FatCow_Icons16x16/track_changes_edit.png",
        encoded: false
    });
    $context_menu = $('<ul id='\"context-menu_dpt_dmsp'></ul>').kendoContextMenu({
        target: $(grid.element).find('#context-menu_but_dpt'),
        showOn: "click",
        open: e => {
            CheckStrRoles('PRODUCT:UPDATE; MSR_PROTOCOL:UPDATE;TECHNOLOG:UPDATE;').then((value) => {
                if (!grid.dataItem(grid.select()) && value == 1) {
                    $(e.item).find('.updt_bc_st').addClass('k-state-disabled');
                } else {
                    let dataItem = grid.dataItem(grid.select());
                    $(e.item).find('.updt_bc_st').removeClass('k-state-disabled');
                };
            });
        },
        select: e => {
            let buttonClass = $(e.item).find("label")[0].className;
            if (buttonClass == 'updt_bc_st') {
                if (grid.dataItem(grid.select())) {
                    let updt_bc_st_wndw;
                    updt_bc_st_wndw = new UpdtBcStWndw();
                    updt_bc_st_wndw.bc_id = grid.dataItem(grid.select()).bc_id;
                    updt_bc_st_wndw.backHandler = (e) => {
                        grid.dataSource.read();
                    };
                    updt_bc_st_wndw.open();
                };
            },
            dataSource: f_lst
        }).data('kendoContextMenu');
    $('ul[data-role="contextmenu"] span').attr('style', 'white-space: nowrap;');
};
var GetAudit = function (element) {

```



```

        that.multiSelectChangePar.treeview.expand($(e.target).closest('li'));
        $(e.target).removeClass('k-i-expand').addClass('k-i-collapse');
    } else {
        that.multiSelectChangePar.treeview.collapse($(e.target).closest('li'));
        $(e.target).removeClass('k-i-collapse').addClass('k-i-expand');
    }
    e.stopPropagation();
});
that.multiSelectChangePar.treeview.wrapper.find('.k-in').off('click').on('click', (e) => {
    that.multiSelectChangePar.treeview.wrapper.find('.k-in').removeClass('k-state-hover')
    let checkbox = $(e.target).siblings('.k-checkbox-wrapper').find('input[type="checkbox"]')
    console.log(checkbox)
    if (checkbox.prop('checked')) {
        checkbox.prop('checked', false)
    } else {
        checkbox.prop('checked', true)
    }
    checkbox.trigger('change')
    e.stopPropagation();
});});});},
schema: {
    model: {
        id: "value",
        children: "items"
    }
},
placeholder: lang('Оберіть пращура'),
checkboxes: {
    template: function (data) {
        // додаємо чекбокси, сховуючи відображення по умові з hide_checkbox
        let kendoOutput = '<input type=\'checkbox\' tabindex=\'-1\' ' +
            ((data.item.enabled === false) ? 'disabled' : '') + ' ' +
            (data.item.checked ? 'checked' : '') + ' id=\'_' + (data.item.uid) + \' class=\'k-checkbox ui_input_tree_checkbox_\' + (data.item.value) +
            (data.item.hide_checkbox ? \'style=\'display: none; margin: -7px 0px 0px 0px;\' ' : \'style=\'margin: -7px 0px 0px 0px;\' ' ) +
            ' /><span class=\'k-checkbox-label checkbox-span\'></span>';
        return kendoOutput;
    },
    open: function (e) {
        // перше відкриття віджету
        if (!e.sender.isFirstOpen) {
            // додаємо задній текст на інпут фільтрації
            //e.sender.filterInput.attr("placeholder", lang('Кінцевий підрозділ містить') + " :");
            // розкриваємо всі дочірні елементи
            e.sender.tree.children().find('.k-icon.k-i-expand:first').trigger('click');
            // трігерим флаг першого відкриття
            e.sender.isFirstOpen = true;
            // додаємо значок звернути/розгорнути всі дочірні елементи
            let expandButton = `<span style="margin-left: -45px;"
                class="k-link k-link-refresh" aria-label="Refresh the Search" title="${lang('згорнути')} + '/' + lang('розгорнути')} + ' ' + lang('всі
дочірні елементи')">`
            <span unselectable="on" class="k-klevis k-icon k-i-expand" style="font: 20px/1 WebComponentsIcons; margin-top: -
2px;"></span></span>`;
            e.sender.filterInput.after(expandButton)
            // додаємо подію згортання (якщо знайшли хоча б 1 розгорнутий елемент не враховуючи перший - загальний) / розгортання всіх
дочірніх елементів
            e.sender.filterInput.siblings('span').children('.k-klevis.k-icon.k-i-expand').off('click').on('click', function (e) {
                if ($(e.target).closest('.k-list-filter').siblings('.k-widget.k-treeview').find('.k-icon.k-i-collapse:not(:first)').length) {
                    $(e.target).closest('.k-list-filter').siblings('.k-widget.k-treeview').find('.k-icon.k-i-collapse:not(:first)').trigger('click');
                } else {
                    $(e.target).closest('.k-list-filter').siblings('.k-widget.k-treeview').find('.k-icon.k-i-expand').trigger('click');
                }
            });
            // наступні відкриття віджету
        } else {
            console.log('second open');
        }
    },
    select: function (e) {
        // забираємо трігер для полів де прихований чекбокс
        if ($(e.node).children('div').find('input').css('display') == 'none') {
            e.preventDefault();
        }
    },
    change: function (e) {
        // якщо обрано більше 1 поля - обираємо останнє та трігерим change
        if (this.value().length > 1) {
            let last_item = this.value().slice(-1)
            e.sender.value([]);
            e.sender.value(last_item)
        }
    }
}

```



```

        e.sender.trigger('change')
    } else {
        // для всіх попередньо виділених жовтим ітемів - забираємо окрас через видалення класу
        e.sender.tree.children().find('span.k-in.k-state-selected').removeClass('k-state-selected')
        // шукаємо щойно чекнутий чекбокс та красим текст біля нього в жовтий (виділення)
        e.sender.tree.children().find('input[type="checkbox"]:checked').closest('span.k-checkbox-wrapper').siblings('span.k-in').addClass('k-state-selected')
    }
},
filtering: function (e) {
    console.log("filtering");
    setTimeout(() => {
        // навішую нормальне поведження на телефоні
        that.multiSelectChangePar.treeview.wrapper.find('.k-checkbox-wrapper').css('margin', '0px 0px 0px 5px')
        that.multiSelectChangePar.treeview.wrapper.find('.k-icon.k-i-expand, .k-icon.k-i-collapse').off('click').on('click', (e) => {
            if ($(e.target).hasClass('k-i-expand')) {
                that.multiSelectChangePar.treeview.expand($(e.target).closest('li'));
                $(e.target).removeClass('k-i-expand').addClass('k-i-collapse');
            } else {
                that.multiSelectChangePar.treeview.collapse($(e.target).closest('li'));
                $(e.target).removeClass('k-i-collapse').addClass('k-i-expand');
            }
        });
        e.stopPropagation();
    });
    //-----
    that.multiSelectChangePar.treeview.wrapper.find('.k-in').off('click').on('click', (e) => {
        that.multiSelectChangePar.treeview.wrapper.find('.k-in').removeClass('k-state-hover')
        let checkbox = $(e.target).siblings('.k-checkbox-wrapper').find('input[type="checkbox"]')
        console.log(checkbox)
        if (checkbox.prop('checked')) {
            checkbox.prop('checked', false)
        } else {
            checkbox.prop('checked', true)
        }
        checkbox.trigger('change')
        e.stopPropagation();
    });
}, 1000);
}
}).data("kendoDropDownTree");
this.dropDownIstDpr = $(this.plg_dpr_name_full).find("#dropDownIstDpr").kendoDropDownList({
    optionLabel: lang("Виберіть підприємство") + "...",
    dataTextField: "dpr_name",
    dataValueField: "dpr_id",
    filter: "startswith",
    value: $v("PO_FIRM"),
    dataSource: {
        transport: {
            read: function (options) {
                apex.server.process("GetFirm",
                    {
                        }, {
                        dataType: "json",
                        success: function (pData) {
                            options.success(pData);
                        }
                    });
            }
        }
    }
}).data("kendoDropDownList");
this.dropDownTp = $(this.plg_ist_tp).find("#dropDownTp").kendoDropDownList({
    highlightFirst: true,
    dataTextField: "name",
    dataValueField: "value",
    minLength: 1,
    suggest: true,
    autoBind: false,
    autoWidth: true,
    dataSource: {
        transport: {
            read: function (options) {
                apex.server.process("GetIstTp", {}, {
                    dataType: "json",
                    success: function (pData) {
                        options.success(pData);
                    }
                });
            }
        }
    },
    change: function (e) {
        }).data("kendoDropDownList");
    }
}

```

// file 'sql ins upd del web'

procedure InsDocPolygon4Wndw(

```

p_doc_id      out document.doc_id%type,
p_doc_obi_ts  out document.doc_obi_ts%type,
p_dct_code    in  doc_type.dct_code%type,
p_param       in  varchar2
)
is
  l_values apex_json.t_values;
  l_doc_num      document.doc_num%type;
  l_doc_dt       document.doc_dt%type;
  l_doc_st       document.doc_st%type;
  l_doc_dpr      document.doc_dpr%type;
  l_dct_tp       doc_type.dct_name%type;
  l_doc_dct      document.doc_dct%type;
  l_doc_spc      document.doc_spc%type;
  l_doc_name     document.doc_name%type;
  l_doci_itm     doc_item.doci_itm%type;
  l_doci_ist_place doc_item.doci_ist_place%type;
  l_dspc_spc     doc_spc.dspc_spc%type;
  l_doci_id      doc_item.doci_id%type;
  l_doci_obi_ts  doc_item.doci_ist_place%type;
  l_dspc_id      doc_spc.dspc_id%type;
  l_dspc_obi_ts  doc_spc.dspc_obi_ts%type;
  l_nms_message  notifymessage.nms_message%type;
  l_wu          document.doc_wu%type;
  l_wu4spc      notifymessage.nms_wu%type;
begin
  l_wu := dis_pltec.dis_pltec_context.GetContext('uid');
  apex_json.parse(p_values => l_values, p_source => p_param);
  l_doc_num      := apex_json.get_varchar2(p_path => 'doc_num', p0 => 1, p_values => l_values);
  l_doc_dt       := apex_json.get_varchar2(p_path => 'doc_dt', p0 => 1, p_values => l_values);
  l_dct_tp       := apex_json.get_varchar2(p_path => 'dct_tp', p0 => 1, p_values => l_values);
  l_doc_st       := apex_json.get_varchar2(p_path => 'doc_st', p0 => 1, p_values => l_values);
  l_doc_dpr      := apex_json.get_varchar2(p_path => 'doc_dpr', p0 => 1, p_values => l_values);
  l_doci_ist_place := apex_json.get_varchar2(p_path => 'doci_ist_place', p0 => 1, p_values => l_values);
  l_doci_itm     := apex_json.get_varchar2(p_path => 'doci_itm', p0 => 1, p_values => l_values);
  l_dspc_spc     := apex_json.get_varchar2(p_path => 'dspc_spc', p0 => 1, p_values => l_values);
  l_doc_name     := apex_json.get_varchar2(p_path => 'doc_name', p0 => 1, p_values => l_values);
  --insert document
  select dct.dct_id
  into l_doc_dct
  from doc_type dct
  where dct.dct_code = p_dct_code;
  document_dm.Insert_Document(
    p_doc_id      => p_doc_id,
    p_doc_obi_ts  => p_doc_obi_ts,
    p_doc_dct     => l_doc_dct,
    p_doc_dpr     => l_doc_dpr,
    p_doc_spc     => null,
    p_doc_ca      => null,
    p_doc_wu      => l_wu,
    p_doc_seria   => null,
    p_doc_num     => null,
    p_doc_dt      => nvl(l_doc_dt, sysdate),
    p_doc_name    => l_doc_name,
    p_doc_st      => nvl(l_doc_st, 'EDIT'),
    p_doc_src     => 'HAND',
    p_doc_num_org => null,
  );
  --create doc_num
  l_doc_num := pltec_common.ProcDocNum(p_doc_id);
  l_wu4spc := pltec_common.GetWu4Spc(p_spc => l_doc_spc);
  whs_dm.Ins_doc_item(
    p_doci_id => l_doci_id,
    p_doci_obi_ts => l_doci_obi_ts,
    p_doci_doc => p_doc_id,
    p_doci_itm => l_doci_itm,
    p_doci_itpt => pltec_common.GetPartZero(p_itm => l_doci_itm, p_date => l_doc_dt, p_st => 'ACCEPT', p_itpt_kind => 'NORMAL'),
    p_doci_ius => null,
    p_doci_count => null,
    p_doci_kind => 'ITM',
    p_doci_weight => null,
    p_doci_descript => null,
    p_doci_comment => null,
    p_doci_ist_place => l_doci_ist_place
  );

  opercard_dm.Ins_Doc_Spc(
    p_dspc_id      => l_dspc_id,
    p_dspc_obi_ts  => l_dspc_obi_ts,

```

```

    p_dspc_doc    => p_doc_id,
    p_dspc_doci   => null,
    p_dspc_spc    => l_dspc_spc,
    p_dspc_jbp    => null,
    p_dspc_jbpa   => null,
    p_dspc_comment => null,
    p_dspc_rcpjbp => null,
    p_dspc_equip_qnt => 1,
    p_dspc_nid     => null,
    p_dspc_dt      => sysdate,
    p_dspc_rel     => 'EXEC'
);
exception
when others then
    raise_application_error(-20000, dis_sys.dis_message_util.get_message(msgCOMMON_EXCEPTION,
'dis_pltec.test_yurchuk.InsDocPolygon4Wndw', chr(10)
                                ||dbms_utility.format_error_stack
                                ||dbms_utility.format_error_backtrace));

end;

procedure UpdDocPolygon4Wndw(
    p_doc_id      in out document.doc_id%type,
    p_doc_obi_ts  in out document.doc_obi_ts%type,
    p_param       in  varchar2
)
is
    l_values apex_json.t_values;
    l_doc_num      document.doc_num%type;
    l_doc_dt       document.doc_dt%type;
    l_doc_st       document.doc_st%type;
    l_doc_dpr      document.doc_dpr%type;
    l_dct_tp       doc_type.dct_name%type;
    l_doc_dct      document.doc_dct%type;
    l_doc_spc      document.doc_spc%type;
    l_doc_name     document.doc_name%type;
    l_doci_itm     doc_item.doci_itm%type;
    l_doci_ist_place doc_item.doci_ist_place%type;
    l_dspc_spc     doc_spc.dspc_spc%type;
    l_doci_id      doc_item.doci_id%type;
    l_doci_obi_ts  doc_item.doci_ist_place%type;
    l_dspc_id      doc_spc.dspc_id%type;
    l_dspc_obi_ts  doc_spc.dspc_obi_ts%type;

    l_nms_message  notifymessage.nms_message%type;
    l_wu           document.doc_wu%type;
    l_wu4spc       notifymessage.nms_wu%type;
    r_doc          document%rowtype;
    r_doci         doc_item%rowtype;
    r_dspc         doc_spc%rowtype;
begin
    l_wu := dis_pltec.dis_pltec_context.GetContext('uid');
    apex_json.parse(p_values => l_values, p_source => p_param);
    l_doc_num      := apex_json.get_varchar2(p_path => 'doc_num', p0 => 1, p_values => l_values);
    l_doc_dt       := apex_json.get_varchar2(p_path => 'doc_dt', p0 => 1, p_values => l_values);
    l_dct_tp       := apex_json.get_varchar2(p_path => 'dct_tp', p0 => 1, p_values => l_values);
    l_doc_st       := apex_json.get_varchar2(p_path => 'doc_st', p0 => 1, p_values => l_values);
    l_doc_dpr      := apex_json.get_varchar2(p_path => 'doc_dpr', p0 => 1, p_values => l_values);
    l_doci_ist_place := apex_json.get_varchar2(p_path => 'doci_ist_place', p0 => 1, p_values => l_values);
    l_doci_itm     := apex_json.get_varchar2(p_path => 'doci_itm', p0 => 1, p_values => l_values);
    l_dspc_spc     := apex_json.get_varchar2(p_path => 'dspc_spc', p0 => 1, p_values => l_values);
    l_doc_name     := apex_json.get_varchar2(p_path => 'doc_name', p0 => 1, p_values => l_values);
    --update document
    select doc.* into r_doc from document doc where doc.doc_id = p_doc_id;
    document_dm.Update_Document(
        p_doc_id      => p_doc_id,
        p_doc_obi_ts  => p_doc_obi_ts,
        p_doc_spc     => r_doc.doc_spc,
        p_doc_dpr     => l_doc_dpr,
        p_doc_ca      => r_doc.doc_ca,
        p_doc_wu      => l_wu,
        p_doc_seria   => r_doc.doc_seria,
        p_doc_num     => r_doc.doc_num,
        p_doc_num_org => r_doc.doc_num_org,
        p_doc_dt_org  => r_doc.doc_dt_org,
        p_doc_dt      => nvl(l_doc_dt, r_doc.doc_dt),
        p_doc_name    => l_doc_name,
        p_doc_st      => r_doc.doc_st,
        p_doc_dpr_node => r_doc.doc_dpr_node,
        p_doc_num_fix => r_doc.doc_num_fix,
        p_doc_priority => r_doc.doc_priority,

```

```

    p_doc_num_1c => r_doc.doc_num_1c
);
select doci.* into r_doci from doc_item doci where doci.doci_doc = p_doc_id;
whs_dm.Upd_doc_item(
    p_doci_id => r_doci.doci_id,
    p_doci_obi_ts => r_doci.doci_obi_ts,
    p_doci_doc => p_doc_id,
    p_doci_itm => l_doci_itm,
    p_doci_itpt => pltec_common.GetPartZero(p_itm => l_doci_itm, p_date => l_doc_dt, p_st => 'ACCEPT', p_itpt_kind => 'NORMAL'),
    p_doci_ius => r_doci.doci_ius,
    p_doci_count => r_doci.doci_count,
    p_doci_kind => 'ITM',
    p_doci_descript => r_doci.doci_descript,
    p_doci_comment => r_doci.doci_comment
);
trace_dm.WriteTrace(p_section => 'test_yurchuk', p_title => 'l_doci_ist_place', p_message => 'l_doci_ist_place-{'||l_doci_ist_place||'}', p_trc_lvl
=> 3);
if l_doci_ist_place is not null then
    select doci.doci_obi_ts
    into l_doci_obi_ts
    from doc_item doci
    where doci.doci_id = r_doci.doci_id;
    whs_dm.Upd_doc_item_column(p_doci_id => r_doci.doci_id,
        p_doci_obi_ts => l_doci_obi_ts,
        p_column_name => 'doci_ist_place',
        p_val => l_doci_ist_place);
end if;
select dspc.* into r_dspc from doc_spc dspc where dspc.dspc_doc = p_doc_id;
opercard_dm.Upd_Doc_Spc(
    p_dspc_id => r_dspc.dspc_id,
    p_dspc_obi_ts => r_dspc.dspc_obi_ts,
    p_dspc_doc => p_doc_id,
    p_dspc_doci => null,
    p_dspc_spc => l_dspc_spc,
    p_dspc_jbp => null,
    p_dspc_jbpa => null,
    p_dspc_comment => null,
    p_dspc_rcpjbp => null,
    p_dspc_equip_qnt => 1,
    p_dspc_nid => null,
    p_dspc_dt => sysdate,
    p_dspc_rel => 'EXEC'
);

exception
when others then
    raise_application_error(-20000, dis_sys.dis_message_util.get_message(msgCOMMON_EXCEPTION,
'dis_pltec.test_yurchuk.UpdDocPolygon4Wndw', chr(10)
||dbms_utility.format_error_stack
||dbms_utility.format_error_backtrace));

end;
procedure DelDocPolygon4Wndw(
    p_doc_id in out document.doc_id%type,
    p_doc_obi_ts in out document.doc_obi_ts%type
)
is
    l_rec_st document.doc_rec_st%type;
begin
    -- статус
    select doc.doc_rec_st into l_rec_st
    from document doc
    where doc.doc_id = p_doc_id
    and doc.doc_obi_ts = p_doc_obi_ts;
    -- лог видалення
    if l_rec_st != dis_const.V_DDN_REC_ST_D then
        document_dm.Del_Document(
            p_doc_id => p_doc_id,
            p_doc_obi_ts => p_doc_obi_ts,
            p_type => 0);
        -- видалення номера
        document_dm.Delete_DocNum(p_doc_id => p_doc_id);
    for cr in (select d2d.d2d_id, d2d.d2d_obi_ts
        from doc2doc d2d
        where d2d_doc_par = p_doc_id
        or d2d_doc_ch = p_doc_id
    )
    loop
        document_dm.Del_Doc2doc(p_d2d_id => cr.d2d_id,
            p_d2d_obi_ts => cr.d2d_obi_ts,

```

```

        p_mode => 0);
end loop;
end if;
-- фіз видалення
if l_rec_st = dis_const.V_DDN_REC_ST_D then
    document_dm.Delete_Document(
        p_doc_id => p_doc_id,
        p_doc_obi_ts => p_doc_obi_ts);
end if;
exception
    when others then
        raise_application_error(-20000, dis_sys.dis_message_util.get_message(msgCOMMON_EXCEPTION, 'test_yurchuk.DelDocPolygon4Wndw',
chr(10)||dbms_utility.format_error_stack || dbms_utility.format_error_backtrace));
end;
apex_json.parse(p_values => l_values, p_source => l_param);
l_doc_id      := apex_json.get_varchar2(p_path => 'doc_id', p0 => 1, p_values => l_values);
l_doc_obi_ts  := apex_json.get_varchar2(p_path => 'doc_obi_ts', p0 => 1, p_values => l_values);
begin

if (p_mode = 'insert') then
    test_yurchuk.InsDocPolygon4Wndw(p_doc_id => l_doc_id,
        p_doc_obi_ts => l_doc_obi_ts,
        p_dct_code => p_dct_code,
        p_param => l_param);
    trace_dm.WriteTrace(p_section => 'test_yurchuk', p_title => 'DocPolygon_WEB', p_message => 'l_doc_id-{'||l_doc_id||'}', p_trc_lvl => 3);
elseif (p_mode = 'update') then
    trace_dm.WriteTrace(p_section => 'test_yurchuk', p_title => 'DocPolygon_WEB', p_message => 'update', p_trc_lvl => 3);
    test_yurchuk.UpdDocPolygon4Wndw(p_doc_id => l_doc_id,
        p_doc_obi_ts => l_doc_obi_ts,
        p_param => l_param);
elseif (p_mode = 'delete') then
    test_yurchuk.DelDocPolygon4Wndw(p_doc_id => l_doc_id,
        p_doc_obi_ts => l_doc_obi_ts);
end if;
--trace_dm.WriteTrace(p_section => 'test_yurchuk', p_title => 'DocPolygon_WEB', p_message => 'end', p_trc_lvl => 3);
apex_json.open_object;
apex_json.write(p_name => 'err', p_value => 0);
apex_json.write(p_name => 'doc_id', p_value => l_doc_id);
apex_json.write(p_name => 'doc_obi_ts', p_value => l_doc_obi_ts);
apex_json.close_object;
exception
    when others then
        apex_json.open_object;
        apex_json.write(p_name => 'err', p_value => 1);
        apex_json.write(p_name => 'err_mess', p_value => sqlerrm);
        apex_json.write(p_name => 'source', p_value => p_param);
        apex_json.close_object;
end;
end;

// tree item storage

declare
    l_sql varchar2(32767);
begin
    l_sql := '
        with
            ist_ as(
                select ist.ist_id
                from dis_pltec.vm_item_storage ist
                where ist.ist_tp = "PLACE" and ist.ist_ist is null and ist.ist_dpr = "||:P1000_FIRM||"
            )
        select level, ist.ist_id as id, ist.ist_ist, case when "||:P1000_LANG||" in ("UKR") then ist.ist_name_ukr_f else ist.ist_name_rus_f end as name,
connect_by_isleaf
        from dis_pltec.vm_item_storage ist
        start with ist.ist_id in (select ist_ist_id from ist_)
        connect by prior ist.ist_id = ist.ist_ist
        order siblings by ist.ist_name';

    dis_pltec.trace_dm.WriteTrace(p_section => 'test_yurchuk', p_title => 'l_sql', p_message => l_sql, p_trc_lvl => 3);
    dis_pltec.json_util.Get_JsonTree4Select_WEB(p_sql => l_sql);
end;

```

Додаток Г (обов'язковий)
 Протокол перевірки на плагіат.

**ПРОТОКОЛ
 ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
 НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: Автоматизована інформаційна система для розвідки родовищ корисних копалин

Тип роботи: магістерська кваліфікаційна робота
 (БДР, МКР)

Підрозділ: кафедра Автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації

(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 99.2% Схожість 0.8%

Аналіз звіту подібності (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Роман МАСЛІЙ
 (підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Марія ЮРЧУК
 (підпис) (прізвище, ініціали)

Керівник роботи _____ Євген ПАЛАМАРЧУК
 (підпис) (прізвище, ініціали)

Додаток Д (додатковий)

Довідка про впровадження впровадження

ДОВІДКА
про впровадження результатів магістерської кваліфікаційної роботи
Юрчук Марії Сергіївни
на тему:

«Автоматизована інформаційна система для розвідки родовищ
корисних копалин»

Зі зростанням потреби у сировинних ресурсах та необхідністю оптимізації видобувних процесів, впровадження сучасних інформаційних технологій стає критично важливим. Використання автоматизованих систем дозволяє значно підвищити точність та ефективність геологорозвідувальних робіт, зменшити витрати та час на обробку даних. Таким чином, розробка автоматизованої інформаційної системи для розвідки родовищ корисних копалин є надзвичайно актуальною в сучасних умовах.

Магістранткою Юрчук М.С. було розроблено автоматизовану інформаційну систему для розвідки родовищ корисних копалин. Система була успішно впроваджена на одному з проєктів товариства з обмеженою відповідальністю «Тестматік».

Метою впровадження є підвищення ефективності процесів розвідки, зменшення витрат і підвищення точності отриманих даних завдяки використанню сучасних інформаційних технологій. Впроваджене рішення базується на використанні технологій Oracle APEX, JavaScript, KendoUI, JQuery та PL/SQL. Основні функції системи включають збір, зберігання та обробку геологічних даних, аналіз даних та формування звітів, інтеграцію з іншими системами та забезпечення безпеки даних.

Цим підтверджуємо, що основні результати магістерської кваліфікаційної роботи Юрчук Марії Сергіївни були успішно впроваджені. Впровадження розробленої системи на проєкті дозволило знизити витрати на проведення розвідувальних робіт та підвищити точність обробки даних, що значно покращило загальну ефективність проєкту. Рекомендується продовжити розвиток системи з розширенням її функціональних можливостей, регулярно оновлювати та оптимізувати алгоритми обробки даних, а також проводити навчання персоналу для ефективного використання системи.

Директор

Владислав КОШМАН