

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "магістр"

на тему:

**Системний аналіз методів та розробка СППР для управління та підбору
об'єктів нерухомості**

Виконав: студент групи 6145 м

_____ ***Золін О.В.***
(підпис)

Керівник роботи:

старший викладач
(посада, науковий ступінь вчене звання)

_____ ***Павленко А.Ю.***
(підпис)

м. Миколаїв – 2023 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

Спеціальність 124 "Системний аналіз"

Освітня програма "Системний аналіз"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

д.ф-м.н, проф. Ушкац М.В.

" ____ " _____ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти "магістр"

Студенту Золіну Олексію Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Системний аналіз методів та розробка СППР для управління та підбору об'єктів нерухомості

Керівник роботи Павленко Альона Юріївна

Затверджені наказом ректора № 1098-уч. від "18" вересня 2023 року.

2. Термін подання роботи: 15 грудня 2023 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали наукового стажування

4. Перелік питань, що належать до розробки (найменування розділів)

Аналіз методів підбору об'єктів нерухомості. Розробка проекту та концепції СППР. Реалізація проекту СППР для управління та підбору об'єктів нерухомості. Розрахунок економічної ефективності впровадження СППР. Охорона праці. Охорона навколишнього середовища.

5. _____ Перелік _____ презентаційних _____ матеріалів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Павленко А.Ю.	05.09.2022	22.12.2022
2	Павленко А.Ю.	23.12.2022	15.06.2023
3	Павленко А.Ю.	15.06.2023	25.09.2023
4	Павленко А.Ю.	25.09.2023	14.11.2023
5	Павленко А.Ю.	25.09.2023	14.11.2023
6	Павленко А.Ю.	25.09.2023	14.11.2023

7. Дата видачі завдання 05 вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Дослідження та аналіз методів підбору об'єктів нерухомості	22.12.2022	
2	Розробка концепції СППР	15.03.2023	
3	Розділ з розробки проектних рішень	15.06.2023	
4	Реалізація проекту та розробка робочої документації	25.09.2023	
5	Обчислення економічної ефективності	14.11.2023	
6	Вирішення питань охорони праці та навколишнього середовища	14.11.2023	
7	Оформлення пояснювальної записки	14.11.2023	
8	Подання магістерської роботи на перевірку на плагіат	16.11.2023	
9	Подання магістерської роботи рецензенту	20.11.2023	
10	Подання магістерської роботи на державний захист	10.12.2023	

Студент

(підпис)

Золін О.В.

(ПІБ)

Керівник роботи

(підпис)

Павленко А.Ю.

(ПІБ)

УДК 004.42

Золін О.В. Системний аналіз методів та розробка СППР для управління та підбору об'єктів нерухомості. Магістерська кваліфікаційна робота зі спеціальності 124 — системний аналіз, освітня програма — системний аналіз. Миколаїв: НУК, 2023. 108 с.

На укр.мові. Бібліогр.: 25 назв; рис.: 40 ; табл. 08.

Магістерська робота присвячена підвищенню ефективності СППР класичної архітектури за рахунок використання сучасних технологій в галузі машинного навчання, штучних нейронних мереж, штучного інтелекту та сучасних архітектурних концепцій побудови програмного забезпечення.

Виконано та описано весь цикл розробки спеціалізованої СППР. Розробка включає в себе розгляд теоретичного базису СППР та аналіз потенційної користі застосування в сфері управління нерухомістю; розробку концептуальної моделі програмного засобу, моделювання, проектування та реалізацію програмного засобу; оцінку економічної ефективності та доцільності розробки. Також висвітлюються аспекти охорони праці та навколишнього середовища.

Ключові слова: система підтримки прийняття рішень, управління портфелем комерційної нерухомості, оперативний та ретроспективний аналіз ринку орендної нерухомості, регресійний аналіз, аналіз часових рядів.

UDC 004.42

Oleksiy V. Zolin. System analysis of methods and development of DSS for management and selection of real estate objects. Master's degree paper in specialty 124 — Systems Analysis, educational program — Systems Analysis. Mykolaiv: NUOS, 2023. 108 p.

In Ukrainian. Bibl.: 25 titles; fig.: 40; tabl.: 08.

The master's thesis is dedicated to enhancing the efficiency of classical architecture DSS (Decision Support Systems) through the use of modern technologies in the field of machine learning, artificial neural networks, artificial intelligence, and contemporary architectural concepts of software development.

The work encompasses and describes the entire cycle of development of a specialized DSS. The development includes examining the theoretical basis of DSS and analyzing potential benefits of its application in the field of real estate management; developing a conceptual model of the software, modeling, designing, and implementing the software; assessing the economic efficiency and feasibility of the development. Also highlighted are aspects of labor protection and environmental safety.

Key words: Decision Support System, management of commercial real estate portfolio, operational and retrospective analysis of the rental real estate market, regression analysis, time series analysis.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	9
ВСТУП	10
1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ МЕТОДІВ ПІДБОРУ ОБ’ЄКТІВ НЕРУХОМОСТІ. ПОСТАНОВКА ЗАДАЧІ	12
1.1. Обумовлення актуальності, постановка проблеми.....	12
1.2. Поняття та класифікація систем СППР	14
1.2.1. Процеси прийняття рішень	14
1.2.2. Загальне поняття про СППР	16
1.2.3. Еволюція СППР	19
1.2.4. Сучасна трансформація СППР	21
1.3. Підходи і методи проектування СППР	23
1.4. Постановка задачі.....	25
Висновки до розділу 1	27
2. РОЗРОБКА ПРОЕКТУ СППР ДЛЯ УПРАВЛІННЯ ТА ПІДБОРУ ОБ’ЄКТІВ НЕРУХОМОСТІ.....	29
2.1. Розробка концепції СППР	29
2.1.1. Поняття концепції програмного забезпечення	29
2.1.2. Математичні моделі проблемного модуля.....	30
2.1.3. Побудова блоків вводу та виводу	35
2.1.4. Вибір методології розробки та управління проектом.....	37
2.2. Загальносистемні рішення	38
2.2.1. Архітектура системи.....	38
2.2.2. Контекстна діаграма системи	40

2.2.3. Аналіз сценаріїв використання.....	41
2.2.4. Функціональні та нефункціональні вимоги до ПЗ	43
2.3. Рішення з інформаційного забезпечення.....	45
2.4. Рішення з програмного забезпечення	48
2.4.1. Розробка діаграми компонентів	48
2.4.2. Розробка діаграми розгортання	49
Висновки до розділу 2	51
3. РЕАЛІЗАЦІЯ ПРОЕКТУ СППР ДЛЯ УПРАВЛІННЯ ТА ПІДБОРУ ОБ'ЄКТІВ НЕРУХОМОСТІ	53
3.1. Вибір інструментів реалізації та розгортання СППР	53
3.2. Розгляд контрольних прикладів	58
3.3. Методика тестування створеного програмного забезпечення	58
3.4. Перевірка відповідності програмного засобу вимогам.....	61
Висновки до розділу 3	63
4. РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ РОЗРОБКИ СППР	65
4.1. Комерційний та технологічний аудит науково-технічної розробки.....	65
4.2. Прогнозування витрат на виконання науково-дослідної (дослідно- конструкторської) роботи.....	69
4.3. Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	74
Висновки до розділу 4	79
5. ОХОРОНА ПРАЦІ	81
5.1. Вступ.....	81
5.2. Аналіз шкідливих та небезпечних факторів.....	82

5.3. Розрахунок системи штучного освітлення приміщення відділу інформаційних технологій	85
5.4. Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів при роботі з персональним комп'ютером	87
5.4.1. Заходи щодо зменшення впливу	87
5.4.2. Техніка безпеки при роботі на комп'ютері	92
Висновки до розділу 5	93
6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	95
6.1. Вступні положення	95
6.2. Забруднення навколишнього середовища	95
6.3. Розробка заходів щодо зменшення забруднення навколишнього середовища.....	98
Висновки до розділу 6	99
ВИСНОВКИ.....	101
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	105
ДОДАТОК А.....	107

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- ARIMA — Autoregressive Integrated Moving Average (Авторегресивна інтегрована ковзна середня)
- CNN — Convolutional Neural Networks (Конволюційні нейронні мережі)
- CRNN — Convolutional Recurrent Neural Networks (Згортково-рекурентні нейронні мережі)
- DNN — Deep Neural Networks (Глибокі нейронні мережі)
- FNN — Feedforward Neural Networks (Фідерні нейронні мережі)
- GRU — Gated Recurrent Unit (Градiєнтне вицілювання)
- HTTP — Hypertext Transfer Protocol (Протокол передачі гіпертексту)
- IDE — Integrated Development Environment (Інтегроване середовище розробки)
- JSON — JavaScript Object Notation (Нотація об'єктів JavaScript)
- LSTM — Long Short-Term Memory (Довго-короткострокова пам'ять)
- MVP — Minimum Viable Product (Мінімально життєздатний продукт)
- ReLU — Rectified Linear Unit (Виправлений лінійний елемент)
- RMSE — Root Mean Square Error (Середньоквадратична помилка)
- RNN — Recurrent Neural Networks (Рекурентні нейронні мережі)
- SQL — Structured Query Language (Мова структурованих запитів)
- UML — Unified Modeling Language (Уніфікована мова моделювання)
- XSS — Cross-Site Scripting (Міжсайтовий скриптинг)
- БД — База Даних

ВСТУП

В даній роботі розглядається створення концепції, розробка та впровадження системи підтримки прийняття рішень (СППР), призначеної для сфери нерухомості, що включає передові технології обробки даних та аналітики. Система, спроектована в класичній архітектурі СППР, інтегрує методи машинного навчання, штучних нейронних мереж і сучасних технологій програмування для оптимізації процесу прийняття рішень в галузі нерухомості.

Робота організована у шість основних розділів. Перший розділ присвячений теоретичному огляду та формулюванню ключових концепцій СППР, з акцентом на їх застосування у сфері нерухомості. У другому розділі представлено концептуальну модель системи, що включає опис її структури та функціоналу. Також розділ описує моделювання та конструювання системи, використовуючи UML для відображення її архітектури. Третій розділ присвячений програмній реалізації системи, зокрема використанню Python, FastAPI, SQLAlchemy та Elasticsearch для створення ефективного та гнучкого програмного забезпечення. Розглядаються аспекти тестування системи, включаючи модульне та функціональне тестування, а також деталі розгортання системи за допомогою Docker та оркестрації. Також в розділі викладено принципи побудови кластеру збору та обробки даних, що використовує технології Apache Spark, Spark Streaming, Hadoop для зберігання та обробки великих обсягів даних. Четвертий розділ роботи містить розрахунок економічної ефективності впровадження СППР в галузі оренди нерухомості. П'ятий та шостий розділи присвячені питанням та нормативам охорони праці та охорони навколишнього середовища.

Мета дослідження: підвищення ефективності СППР класичної архітектури за рахунок використання сучасних технологій в галузі машинного навчання, штучних нейронних мереж та штучного інтелекту.

Об'єкт дослідження: процес прийняття рішень в сфері орендної нерухомості.

Предмет дослідження: система підтримки прийняття рішень в сфері орендної нерухомості, що використовує для вирішення задач аналітичний підхід та моделі машинного навчання, а також штучний інтелект.

Методи дослідження: статистичний аналіз даних, розробка алгоритмів, програмування, тестування програмного забезпечення.

Наукова новизна роботи полягає в розширенні функціональності та підвищенні продуктивності СППР в сфері нерухомості за рахунок використання сучасних моделей аналізу інформації та прогнозування. Створені модулі системи враховують сучасні виклики та тенденції у розробці: потоковий збір даних, кластер зберігання великих даних, REST-API для взаємодії з іншими додатками та сервісами, контейнеризацію для зручного деплоюменту та безперервної інтеграції.

Практичне значення одержаних результатів: розроблене програмне забезпечення сприятиме підвищенню ефективності рішень в сфері орендної нерухомості, забезпечить більш ефективне управління ресурсами операторів оренди нерухомості.

Апробація результатів магістерської кваліфікаційної роботи:

1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ МЕТОДІВ ПІДБОРУ ОБ'ЄКТІВ НЕРУХОМОСТІ. ПОСТАНОВКА ЗАДАЧІ

1.1. Обумовлення актуальності, постановка проблеми

Системи підтримки прийняття рішень (СППР) є важливим інструментом у сучасному бізнес-менеджменті, що допомагає в оптимізації процесів прийняття рішень через ефективний аналіз даних. СППР можна визначити як комп'ютеризовані системи, що асистують у прийнятті рішень за допомогою збору, обробки та візуалізації великих обсягів даних. Вони забезпечують детальний аналіз складних інформаційних наборів та відіграють ключову роль у виявленні трендів, виявленні аномалій та вирішенні проблем.

Важливість СППР полягає у їх здатності перетворювати сиру інформацію у зрозумілі, аналітичні висновки дозволяють керівникам приймати обґрунтовані рішення. Це включає оцінку ризиків, прогнозування результатів, планування ресурсів та оптимізацію процесів. Особливо велике значення СППР мають у сферах, де потрібно швидко реагувати на зміни ринкових умов або виявляти можливості для інновацій.

У сфері нерухомості впровадження СППР є особливо актуальним, оскільки ринок нерухомості характеризується високою динамікою та складністю. Системи підтримки прийняття рішень можуть надати величезну користь, допомагаючи аналізувати великі масиви даних щодо цін, локацій, ринкових тенденцій, історії продажів, законодавчих змін та багато іншого.

Ключовими аспектами СППР для сфери нерухомості є:

1. Аналіз ринку нерухомості - ефективне використання СППР для збору, обробки та аналізу даних про ринок нерухомості. Це дозволяє аналізувати цінові тенденції, визначати потенційно привабливі інвестиційні можливості та прогнозувати майбутні зміни вартості об'єктів.

2. Управління портфелем нерухомості - централізований збір та управління інформацією про всі об'єкти нерухомості, з якими працює

компанія. СППР може забезпечити оновлену інформацію про статус кожного об'єкта, його історію угод та поточні оцінки вартості.

3. Прогнозування та моделювання - використання передових алгоритмів машинного навчання та аналітичних моделей для прогнозування розвитку ринкових ситуацій, а також для оцінки впливу різних факторів на вартість нерухомості.

4. Візуалізація даних - розробка інтуїтивно зрозумілих інструментів для візуалізації даних, що дозволять користувачам легко оцінювати складні набори даних, отримувати важливу інформацію на один погляд та приймати обґрунтовані рішення.

Системи підтримки прийняття рішень у сфері нерухомості виконують критично важливу функцію у зборі, аналізі та прогнозуванні вартості нерухомості, як на поточний момент, так і в динаміці. Ці системи важливі не лише для кінцевих споживачів, але й для інвесторів, оскільки вони допомагають в оцінці привабливості ринкових пропозицій. Для інвесторів, здатність прогнозувати динаміку вартості та формувати справедливую ціну є вирішальною не тільки при прийнятті рішення про покупку або оренду об'єкта з точки зору привабливості інвестицій, але й у контексті управління поточними активами. Важливим аспектом для інвестора є також здатність своєчасно відмовитися від активів, які втрачають свою привабливість на ринку.

Центральним елементом такої системи є інструмент, що надає можливість проведення аналізу та прогнозування реальної ринкової вартості та орендних ставок об'єктів нерухомості. Враховуючи обмеження в часі та ресурсах, основна мета полягає у розробці, проектуванні та реалізації програмного засобу, який би дозволив перевірити та підтвердити запропоновані теоретичні концепції. Фокус роботи спрямовано на прогнозування актуальної вартості житлової та комерційної нерухомості, а також на створення індикаторів доцільності збільшення чи зменшення

кількості об'єктів певної категорії в портфелі, виходячи з аналізу їхньої потенційної доходності в середньостроковій перспективі.

1.2. Поняття та класифікація систем СППР

1.2.1. Процеси прийняття рішень

Рішення можна визначити як виважений набір дій, запроваджених особою, що приймає рішення, з метою досягнення певного бажаного стану або цілі об'єкта чи системи управління. Цей процес включає розумову діяльність та вияв волі людини. Основними атрибутами рішення є наявність можливості вибору серед декількох альтернативних варіантів, оскільки без альтернатив вибір та відповідно рішення є неможливими; наявність чітко визначеної мети, адже вибір, що не має цілі, не розглядається як рішення; а також необхідність вольового втручання особи, яка приймає рішення, оскільки саме ця особа формує рішення, вирішуючи між різними мотивами та думками.

Прийняття рішення є процесом вибору найбільш придатного варіанту з доступних альтернатив або ранжування цих альтернатив. Цей процес базується на глибокому розумінні об'єкта управління, його поточних та потенційних майбутніх процесів, а також на наявності критеріїв для оцінки ефективності та якості обраного рішення. Таким чином, необхідною умовою є наявність точної моделі об'єкта та моделі прийняття та оцінки рішення. Модель прийняття рішень охоплює формальне представлення поставленої задачі та процесу прийняття рішення.

Теорія прийняття рішень (ТПР), яка почала розвиватися ще з XVIII століття, займається фундаментальними питаннями, такими як визначення критеріїв оптимальності. Наукові основи ТПР були закладені під час Другої світової війни, зокрема завдяки роботі Джона фон Неймана та Оскара Моргенштерна, які у 1944 році опублікували книгу з теорії ігор. Задачі, що включають опис і аналіз типів вибору та таких теоретичних конструкцій як "корисність" та "перевага", були ключовими в цій галузі. До середини XX

століття були розроблені теорії оптимізаційного вибору, засновані на бінарних відношеннях переваг. Сучасні моделі базуються на припущеннях щодо того, як індивідууми та колективи здійснюють вибір серед варіантів.

Класифікацію видів рішень представлено в таблиці 1.1.

Таблиця 1.1 – Класифікація видів рішень.

Ознака	Вид рішення		
Ступінь структуризації проблеми	Гарно структуроване	Погано структуроване	Не структуроване
Кількість етапів реалізації рішення	Статичне (один етап)		Динамічне
Рівень інформованості про стан проблеми	Умови визначеності	Умови ризику	Умови невизначеності
Кількість ОНР	Одна особа		Багато осіб
Зміст рішення	Стратегічне		Тактичне

Будь-який процес прийняття рішення здійснюється в декілька основних етапів.

Фаза формулювання задачі охоплює аналіз та діагностування існуючої проблеми, а також уточнення цілей, яких потрібно досягнути з використанням рішення. На цьому етапі здійснюється ідентифікація та опис проблемної ситуації, а також збір необхідної інформації та даних. Визначення конкретної мети рішення допомагає направляти пошук потенційних рішень та відкидати ті, які не відповідають встановленій меті.

Фаза формування рішень включає в себе процеси встановлення обмежень та критеріїв, необхідних для прийняття рішень, а також розробку альтернативних варіантів рішення. На цьому етапі здійснюється визначення обмежень, які допомагають відсіяти непридатні варіанти, та критеріїв, які полегшують вибір найкращих рішень серед прийнятних. Далі відбувається процес вибору допустимих альтернатив, який полягає у виявленні та розробці різних можливих варіантів рішень.

Фаза вибору рішення охоплює процес оцінки альтернатив та здійснення кінцевого вибору. На цьому завершальному етапі проводиться оцінка різних варіантів з допустимих альтернатив на основі встановлених критеріїв, після чого здійснюється остаточний вибір. Вартість кожної альтернативи зазвичай варіюється, і у випадку недостатньо вираженої переваги одного варіанту над іншими можуть виникнути певні складнощі.

В процесі прийняття рішень особою, що приймає рішення (ОПР), часто виникають помилки, серед яких найбільш поширені є прийняття однобічного рішення, відсутність системного підходу, перевага "звичній" альтернативі, зосередження тільки на позитивних варіантах із ігноруванням потенційних ризиків, емоційно зумовлене рішення, імпульсивний вибір, поспіх, прийняття рішень на основі припущень, бажань або невірних передумов, неправильне тлумачення фактів та неактуальність рішення, коли воно реалізоване невірно або несвоєчасно.

1.2.2. Загальне поняття про СППР

Концепція систем підтримки прийняття рішень (СППР), або Decision Support Systems (DSS), була започаткована в 1970-х роках Горрі та Мортонем. Початкові версії цих систем мало відрізнялися від традиційних управлінських інформаційних систем, що спонукало до використання альтернативної термінології – «системи управлінських рішень».

У науковій літературі існує велика різноманітність визначень СППР, відображаючи їх багатогранність. Наприклад, їх описують як інтерактивні прикладні системи, що забезпечують користувачам, відповідальним за прийняття рішень, легкий доступ до даних і моделей для ефективного вирішення напівструктурованих і неструктурованих задач. Інші визначення вказують на використання СППР для підтримки менеджерів у прийнятті рішень через моделі та процедури обробки даних.

СППР, за одним з визначень, є комп'ютерними інформаційними системами, призначеними для підтримки різноманітних видів діяльності у

складних ситуаціях прийняття рішень. Вони часто використовуються в ситуаціях, де автоматизоване прийняття рішень неможливе або недоречне.

Основними користувачами СППР є менеджери та фахівці. Системи можуть бути використані ними для особистих цілей або як засіб посередництва в управлінні. Значну роль у СППР відіграє інтерфейсне програмне забезпечення, що робить систему інтуїтивно зрозумілою та зручною у використанні. Серцевиною СППР є моделі аналізу даних, які підтримують або доповнюють процес прийняття рішень.

Складність завдань, які вирішуються за допомогою СППР, полягає в тому, що часто немає ясності щодо мети або методів досягнення. СППР не виробляють рішення самостійно, але забезпечують користувачів необхідною інформацією для оцінки та вибору оптимального рішення.

До категорії СППР відносяться системи, які перевищують рамки онлайн-процесів оброблення транзакцій (OLTP). Це можуть бути програми, встановлені на персональних комп'ютерах, що допомагають менеджерам у прийнятті рішень. Сюди також входять виконавчі інформаційні системи, геоінформаційні системи, системи інтерактивного аналізу даних (OLAP), сховища даних, системи виявлення знань та системи підтримки групового прийняття рішень.

Системи підтримки прийняття рішень (СППР) класифікуються за двома основними категоріями: корпоративні (або Enterprise-wide) та настільні (desktop) СППР.

Корпоративні СППР характеризуються їх масштабістю та впровадженням на рівні цілих підприємств. Вони інтегровані з обширними базами даних і використовуються безліччю менеджерів у різних компаніях. Натомість настільні СППР призначені для окремих користувачів і представляють собою компактні системи, встановлені на персональних комп'ютерах менеджерів.

Розвиток корпоративних СППР зумовлений підвищеною увагою до раціонального і взаємопов'язаного управління на рівні підприємства. Ці

системи, забезпечені великими сховищами даних, надають керівникам доступ до широкого спектру інформації про компанію в необхідний час. Корпоративні СППР спрощують процес прийняття рішень, варіюючи від базових систем, що забезпечують загальну інформацію, до складних аналітичних і виконавчих інформаційних систем.

Основні функції корпоративних СППР включають детальну обробку інформації, відокремлення окремих сегментів даних для детального аналізу, а також візуалізацію корпоративних та зовнішніх даних у наочному (графічному та схематичному) вигляді.

У більшості організацій існує взаємодія між корпоративними та настільними СППР, а також між сховищами даних. Розуміння місцезнаходження, методів аналізу та презентації різноманітної інформації є ключовим для ефективного використання цих систем.

Архітектура систем підтримки прийняття рішень (СППР) може бути адаптована як для загального використання на рівні підприємства, так і для індивідуальних потреб окремих користувачів. В контексті використання на рівні підприємства, СППР структуровані таким чином, щоб забезпечувати послідовність у передачі та аналізі даних, починаючи з настільних комп'ютерів користувачів, через мережеві засоби СППР, до серверних систем. У цьому випадку, СППР та бази даних можуть бути розміщені у різних локаціях в межах організації.

Настільні СППР, розраховані на індивідуальних користувачів, хоча і не отримали такої широкої популярності, як корпоративні системи, все ж відіграють свою роль. Вони часто включають використання програмних пакетів для електронних таблиць для здійснення необхідного аналізу на рівні окремого робочого місця. Це може також охоплювати розробку спеціалізованих функцій СППР, які відповідають конкретним вимогам індивідуальних менеджерів.

Підсумуємо ключові аспекти систем підтримки прийняття рішень (СППР) та їх орієнтацію:

- СППР спрямовані на допомогу менеджерам у знаходженні відповідей на питання, пов'язані з прийняттям рішень;
- ці системи є комп'ютеризованими і взаємодіють з іншими системами, щоб допомагати менеджерам у пошуку, розрахунках та аналізі даних, які стосуються прийняття рішень.

Типи СППР включають:

- корпоративна СППР, яка підтримує роботу великої групи менеджерів у єдиній системі з обслуговування клієнтів і має спеціалізоване сховище даних;
- настільна СППР, призначена для індивідуального користувача і встановлена на персональному комп'ютері менеджера.

СППР можуть бути зорієнтовані на різні аспекти, включаючи дані, моделі, комунікації та інше.

1.2.3. Еволюція СППР

Розвиток систем підтримки прийняття рішень (СППР) розпочався у ранні періоди епохи розподіленого обчислення. Їхня історія бере свій початок з 1967 року, і з того часу СППР зазнали значного еволюційного процесу. Цей розвиток включає різні аспекти, такі як модельно-орієнтовані СППР, засоби для формування запитів та звітності, системи аналізу бізнес-даних, оперативне аналітичне оброблення даних (OLAP), групові СППР та виконавчі інформаційні системи.

Теорія розвитку систем підтримки прийняття рішень (СППР) почала формуватися у кінці 1970-х років. В цей період важливими були ідеї розподілу та обміну інформацією. Пітер Кін та Майкл Скотт Мортон запропонували концепції аналізу, проектування, створення, оцінки та розвитку систем підтримки прийняття рішень. На початку 1980-х років, Джон Рокарт і Стівен Алтер опублікували результати своїх досліджень, зосереджені на використанні моделей для підтримки процесу прийняття рішень.

У 1981 році Бончек, Холсапл і Вінстон представили теоретичну рамку для розуміння аспектів, пов'язаних з проектуванням СППР, орієнтованих на знання. Вони визначили чотири ключові компоненти для всіх СППР:

- мовну систему (LS) для прийому повідомлень;
- систему подання (PS) для видачі повідомлень;
- систему знань (KS) для накопичення та зберігання знань;
- систему оброблення проблем (PPS) — «проблемний процесор», що намагається розпізнати і розв'язати проблеми протягом використання СППР

У 1982 році Ральф Спраге та Ерік Карлсон запропонували практичний та зрозумілий огляд того, як організації можуть і повинні будувати СППР. Сучасні підходи до розробки та дослідження СППР включають забезпечення необхідними інструментальними засобами та дослідженнями управління даними, розробку математичних моделей для використання в СППР, а також дослідження поведінки менеджерів при прийнятті рішень.

До кінця 1970-х років розвиток СППР набув нових обрисів з розробкою інтерактивних інформаційних систем, які використовували дані та моделі для допомоги менеджерам в аналізі слабо структурованих проблем. Ці системи стали відомими як системи підтримки прийняття рішень і були визнані ефективними для підтримки прийняття рішень на будь-якому рівні в організації, від окремих операцій до стратегічного планування.

У середині 1980-х розробили новий тип програмного забезпечення для підтримки групового прийняття рішень. З 1990-х років сховища даних та OLAP (інтерактивне аналітичне оброблення) сприяли появі нових типів СППР, орієнтованих на дані.

У ранніх 1990-х роках відбулося значне технологічне просування у сфері систем підтримки прийняття рішень (СППР), що включало перехід від основних комп'ютерних систем (mainframe) до клієнт-серверних архітектур. Цей перехід був спричинений появою об'єктно-орієнтованих технологій, які

продавці впроваджували для створення більш гнучких і масштабованих систем підтримки прийняття рішень, здатних до багаторазового використання. В результаті багато компаній почали модернізувати свої мережеві інфраструктури для адаптації до нових вимог.

Хоча історія систем підтримки прийняття рішень відносно коротка, концепції та технології в цій сфері продовжують інтенсивний розвиток. Поява та швидке поширення Інтернету та Веб-технологій значно прискорили цей розвиток, відкриваючи нові можливості для систем підтримки прийняття рішень та розширюючи їх функціональність.

1.2.4. Сучасна трансформація СППР

Основні тренди розвитку систем підтримки прийняття рішень (СППР) у ХХІ столітті відображають значний технологічний поступ, зумовлений декількома ключовими факторами. Впровадження широкосмугового інтернету відіграло важливу роль у цьому процесі, забезпечуючи швидкий доступ до великих обсягів даних та підвищуючи мобільність та зручність використання систем на різних пристроях. Це, у свою чергу, зробило СППР більш доступними і ефективними для широкого кола користувачів, а також створило можливість збору великої кількості неструктурованих даних, так званих великих даних, BigData.

Цей термін був введений редактором журналу Nature Кліффордом Лінчем 2008 року в спецвипуску, присвяченому вибуховому зростанню світових обсягів інформації. На думку редактора, до великих даних слід відносити потоки даних понад 100 Гб на день.

Спочатку під цим простим терміном були приховані всього лише два поняття - зберігання та обробка даних. Під обробкою даних розуміємо різні методи, моделі, підходи інструменти обробки даних. Пізніше BigData переросло в соціально-економічний феномен, пов'язаний з появою нових технологічних можливостей аналізу великого обсягу даних.

Хмарні технології також суттєво вплинули на розвиток СППР, пропонуючи більш гнучкі та масштабовані рішення. Вони знизили вартість впровадження та обслуговування цих систем, дозволивши навіть малим і середнім підприємствам використовувати передові аналітичні інструменти. Хмарні СППР дозволяють користувачам отримувати доступ до даних і аналітичних інструментів в будь-який час і з будь-якого місця, що підвищує гнучкість і оперативність прийняття рішень.

Машинне навчання та штучні нейронні мережі стали ще одними каталізаторами трансформації СППР. Завдяки здатності обробляти великі масиви даних та робити прогнози або рекомендації на їх основі, ці технології значно підвищили точність та ефективність систем підтримки прийняття рішень. Машинне навчання дозволяє СППР самонавчатися з часом, постійно покращуючи якість прийнятих рішень. Штучні нейронні мережі застосовуються для виявлення складних залежностей у даних, що було б неможливим для обчислення традиційними аналітичними методами.

Сучасні мовні моделі, як компонент систем підтримки прийняття рішень (СППР), можуть ефективно адаптуватися для реалізації мовної системи (LS), що відповідає за прийом повідомлень. Мовні моделі, оснащені розширеними можливостями обробки природної мови, здатні аналізувати запити користувачів, розуміти контекст та навіть вловлювати нюанси у формулюваннях. Це дозволяє системі точно інтерпретувати запити, структуровані як у формальній, так і в неформальній мові. Такий підхід значно покращує інтерактивність і доступність СППР, дозволяючи користувачам формулювати запити більш природним і звичним способом.

У той же час, мовні моделі можуть бути адаптовані для реалізації системи подання (PS) для видачі повідомлень. Це означає, що система здатна не тільки приймати інформацію, але й ефективно комунікувати з користувачем, надаючи відповіді та звіти в зрозумілій, структурованій та зручній для сприйняття формі. Використання мовних моделей для генерації тексту дозволяє СППР надавати детальні пояснення, візуалізації даних, а

також структуровані висновки на основі аналізу, що робить процес прийняття рішень більш прозорим та обґрунтованим для користувачів.

В цілому, розвиток СППР у ХХІ столітті характеризується підвищенням масштабованості, гнучкості, доступності, а також значним поліпшенням аналітичної потужності та точності систем. Таким чином, сучасні СППР відкриті до впровадження великої кількості технологій пошуку, агрегування та зберігання даних, машинного навчання та штучного інтелекту. Також можна вважати, що розвиток технологій, швидкості широкосмугового інтернету, розмірів носіїв даних та швидкості обчислень має вплив, що схожий із впливом на велику кількість інших галузей: технології та інструменти, що двадцять років тому були коштовними та доступними виключно великим корпоративним користувачам, на сьогодні є доступними для широкого кола користувачів за незрівнянно меншою вартістю.

1.3. Підходи і методи проектування СППР.

Загальний процес проектування СППР може бути розділений на три основні етапи:

1. Декомпозиція процесу прийняття рішення. Цей етап передбачає розбивку процесу на елементарні операції та детальний опис цих процесів. Це включає в себе виявлення обмежень, які потрібно подолати з допомогою СППР, та визначення набору алгоритмів для ефективного прийняття рішень.

2. Функціональне проектування СППР. На цьому етапі проводиться аналіз конкретної задачі ухвалення рішень та розробка СППР на функціональному рівні. Це включає в себе аналіз перешкод, що виникають при рішенні задач без використання комп'ютерів, та визначення необхідних функцій, які можна автоматизувати.

3. Специфікація та реалізація системи. Останнім етапом є докладна специфікація функцій системи, її реалізація, тестування та супроводження. На цьому етапі встановлюється відповідність між потрібними функціями та

методами, що підтримують їх реалізацію, та інтеграція цих елементів у загальну структуру СППР.

В процесі проектування важливо врахувати й поширені помилки при прийнятті рішень, що були розглянуті в попередньому розділі.

Оскільки процес побудови СППР починається саме з декомпозиції процесу прийняття рішень людиною (ОПР), в поточному підрозділі доцільно розглянути також стратегії прийняття рішень. В залежності від стратегії, що використовує ОПР, будемо мати різні декомпозиції процесу прийняття рішень. Також формування переліку стратегій дозволить пізніше зробити відбір стратегій, що буде використовувати проектована система.

Оптимізаційна стратегія включає використання математичних моделей та критеріїв оптимізації для вибору найкращої альтернативи. Це дозволяє розглядати різні сценарії та знаходити найефективніші рішення, але може бути обмеженою через якісні критерії та складність обчислень.

Стратегія вибору першої прийнятної альтернативи полягає у виборі першої опції, яка задовольняє базові критерії, без спроби знайти оптимальне рішення. Цей метод ефективний для швидкого вирішення проблеми, але може не забезпечити найкращого результату.

Стратегія аспектного виключення виключає альтернативи, які не відповідають окремим критеріям. Це дозволяє систематично зменшувати кількість варіантів, але може призвести до пропуску потенційно ефективних альтернатив.

Інкрементна стратегія ґрунтується на поступовому поліпшенні поточної ситуації шляхом невеликих змін. Це дає можливість адаптивної реакції на зміни, але може не враховувати більші стратегічні можливості.

Стратегія змішаного сканування передбачає збір та аналіз інформації для відкидання неприйнятних альтернатив, залишаючи кращі варіанти. Це забезпечує більш всебічний аналіз, але може потребувати більше часу.

Аналітико-ієрархічна стратегія включає декомпозицію задачі на ієрархію критеріїв та альтернатив, що дозволяє оцінити різні аспекти рішення та знайти збалансоване рішення. Однак, цей метод може бути складним та часомістким при збільшенні кількості критеріїв та альтернатив.

Наведене формулювання підходів та методів проектування СППР стосується саме ключового блоку системи – «проблемного процесора», але як ми побачили в попередньому підрозділі, є й інші компоненти системи, які відповідають за введення запитів, формування відповідей, а також збору, накопичення, форматування й зберігання даних.

Компоненти введення запитів та формування відповідей системи можуть бути реалізовані, як було визначено раніше, з використанням мовних моделей, метою яких є перетворення довільного текстового запиту в один з стандартних типів запитів до системи, й навпаки, перетворення створеної алгоритмічної відповіді у текст на природній мові.

Задача збору, накопичення, форматування та зберігання даних є технічною задачею організації кластеру потокового збору та зберігання даних, для якої можна використати одне з існуючих сучасних рішень, провівши необхідну адаптацію під потреби задачі, що розглядається.

1.4. Постановка задачі

Відповідно до міркувань, наведених в підрозділі 1.1, інвестори в галузі нерухомості мають потреби в підтримці актуальності вартості оренди, якими вони володіють та надають в оренду, також є затребуваною підтримка в прийнятті рішень при виборі нових об'єктів інвестицій та оцінці доцільності реалізації об'єктів, що знаходяться в портфелі інвестора, через зниження доходності від їх оренди сьогодні або в найближчий перспективі.

Система підтримки прийняття рішень має складатись з чотирьох блоків, один з яких – «проблемний процесор» має бути побудований на базі аналітичної стратегії прийняття рішень та використовувати в роботі сучасні моделі машинного навчання та штучного інтелекту.

Модулі вводу-виводу мають використовувати технології та моделі обробки природньої мови для перетворення запитів на природній мові в запити фіксованої структури, що використовуються системою, та перетворення відповідей проблемного процесору в текст знов таки на природній мові.

Модуль зберігання та обробки даних має використовувати сучасну структуру та технології для потокового збору, структурування та зберігання даних, також необхідне визначенні механізмів збору первинної інформації в механістичний спосіб.

Таким чином, в рамках виконання задачі розробки сучасної системи підтримки прийняття рішень необхідно виконати наступні завдання:

1. Окреслення кола проблем, з яким працює проблемний процесор.
2. Вивчення моделей машинного навчання та штучних нейромереж, що є доречними для розв'язання задач або підзадач проблемного процесора. Обрання тих моделей, що будуть використані для побудови проблемного процесора.
3. Визначення форматів вводу, виводу проблемного процесора, а також типів та структури інформації, що має використовуватись для генерації рішень проблемним процесором.
4. Вивчення та обрання моделей обробки природньої мови для забезпечення відповідності між запитами/відповідями на природній мові та форматом запитів/відповідей проблемного процесора системи.
5. Розробка архітектури збору, агрегації, очищення, форматування та зберігання даних, необхідних для функціонування системи, що розробляється.
6. Розробка концептуальної моделі системи, моделювання та проектування систему.
7. Програмна реалізація розробленої моделі системи та її розгортання.

8. Функціональне тестування системи та перевірка на відповідність системи сформульованим під час моделювання функціональним та нефункціональним вимогам.

Висновки до розділу 1

Перший розділ досліджує та аналізує методи підбору об'єктів нерухомості з використанням систем підтримки прийняття рішень (СППР). Він розкриває значення СППР у сучасному бізнес-менеджменті, особливо в контексті ринку нерухомості. Розділ наголошує на важливості СППР у перетворенні великих обсягів даних на зрозумілі аналітичні висновки, які допомагають керівникам, менеджерам та пересічним громадянам, що мають інтереси в сфері нерухомості, приймати обґрунтовані рішення. СППР в нерухомості забезпечують аналіз цін, локацій, ринкових тенденцій, історії продажів, законодавчих змін тощо, підкреслюючи їх роль у прогнозуванні вартості та інвестиційної привабливості об'єктів.

У цьому контексті, розглядаються ключові аспекти СППР для сфери нерухомості, включаючи аналіз ринку, управління портфелем нерухомості, прогнозування та моделювання, а також візуалізацію даних. Ці системи описані як критично важливі для збору, аналізу та прогнозування вартості нерухомості, відіграючи ключову роль для інвесторів та кінцевих споживачів. Особлива увага приділяється здатності СППР прогнозувати динаміку вартості та визначати справедливую ціну, що є важливим для прийняття рішень стосовно покупки або оренди, а також для управління активами.

Також розглядається теорія та історія розвитку СППР, включаючи їх класифікацію, основні функції та архітектуру. Вони вказують на еволюцію СППР від простих систем до складних інтегрованих рішень, що використовують передові технології, такі як машинне навчання та штучні нейронні мережі. Ця еволюція підкреслює зростаючу важливість СППР у підтримці прийняття рішень на всіх рівнях управління.

Загалом, розділ висвітлює значення та розвиток СППР у контексті ринку нерухомості, підкреслюючи їх здатність покращувати процес прийняття рішень шляхом надання точних аналітичних даних та передбачень. Це допомагає керівникам оптимізувати стратегії та покращувати рішення, пов'язані з інвестиціями та управлінням нерухомістю.

2. РОЗРОБКА ПРОЕКТУ СППР ДЛЯ УПРАВЛІННЯ ТА ПІДБОРУ ОБ'ЄКТІВ НЕРУХОМОСТІ

2.1. Розробка концепції СППР

2.1.1. Поняття концепції програмного забезпечення

Якщо ви не знаєте, в якому напрямку розвивати проєкт, то він навряд чи вибере потрібний шлях самостійно.

Стів Макконнел

Концепція - основний документ проєктів з розробки програмного забезпечення, який буде доступний нетехнічним фахівцям усередині і поза командою. Він має давати можливість замовнику, топ-менеджменту, фахівцям з маркетингу, потенційним споживачам продукту або послуги оцінити розроблювану систему. Як правило, концепція не пишеться технічною мовою, але може включати окремі типи діаграм UML, що не є суто технічними й можуть бути зрозумілими нетехнічним спеціалістам. До таких діаграм можна віднести діаграму компонентів, діаграму сценаріїв використання та деякі діаграми діяльності. Останні мають відноситись до ключових процесів системи й не повинні бути занадто деталізованими. Мета побудови діаграм під час розробки концепції проєкту - візуалізація системи на високому рівні абстракції, що зрозумілий нетехнічним спеціалістам.

Якщо коротко, то етап розробки концепції в життєвому циклі розробки програмного забезпечення є етапом, на якому обираються та описуються ключові моделі та технології, що будуть використані в проєкті для забезпечення необхідної функціональності, описується логічна та фізична архітектура системи на найвищому рівні абстракції.

В поточному підрозділі зробимо вибір необхідних для реалізації проблемного модуля системи моделей машинного навчання та штучних нейронних мереж, а також визначимо підходи для застосування мовних

моделей в блоках вводу – виводу. Інші аспекти розробки концептуальної та функціональної моделі системи будуть показані далі.

2.1.2. Математичні моделі проблемного модуля

Як було сформульовано в попередньому розділі, метою проблемного модуля системи є прогнозування справедливої, збалансованої ціни на об'єкти нерухомості, що необхідно для підтримки рішень інвесторів під час купівлі та продажу об'єктів нерухомості, а також прогнозування цін на нерухомість в динаміці. З цією метою можуть бути використані різні підходи, що схематично показані на рис. 2.1.



Рисунок 2.1 – Сегментація математичних моделей для програмного модуля

Моделі ARIMA та RNN дозволяють обробляти часові залежності, інші моделі на діаграмі працюють з регресійними залежностями, при цьому можливе вивчення залежностей як від чисельних, так й від якісних ознак.

В якості регресійної моделі машинного навчання, як показано вище, розглядатимемо алгоритм підвищення градієнтів, зокрема градієнтне підсилення на деревах рішень. Основна ідея полягає у побудові ансамблю послідовних моделей, де кожна наступна модель виправляє помилки попередньої.

Нехай y_i є істинним значенням, а $\hat{y}_i^{(t)}$ - прогнозованим значенням моделі на t -му кроці. Ціль алгоритму — мінімізувати різницю між цими значеннями.

Функція втрати $L(\theta)$ визначає різницю між фактичними та прогнозованими значеннями. Для регресії часто використовується квадратична функція втрат, яка виражається формулою (2.1)

$$L(\theta) = \sum_{i=1}^n (y_i - \hat{y}_i^{(t)})^2, \quad (2.1)$$

де n — кількість спостережень.

Модель градієнтного підсилення конструюється послідовно. На кожному кроці t будується нове дерево рішень f_t , яке прогнозує градієнт функції втрати по відношенню до попереднього прогнозу.

Прогноз на t -му кроці розраховується за формулою (2.2)

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta \cdot f_t(x_i), \quad (2.2)$$

де η - швидкість навчання, яка контролює внесок кожного дерева.

На кожному кроці виконується оптимізація для визначення оптимальних параметрів дерева f_t .

Алгоритм XGBRegressor включає додаткові параметри регуляризації для запобігання перенавчанню. Регуляризація в XGBoost включає в себе два типи регуляризації:

Регуляризація L1 (ласо) додає штраф до абсолютних значень ваг.

Регуляризація L2 (гребінь) додає штраф до квадратів ваг.

Ці регуляризатори додаються до функції втрат, забезпечуючи баланс між здатністю моделі вловлювати складні закономірності в даних та запобіганням перенавчанню.

Альтернативою регресійним моделям машинного навчання може бути повнозв'язна нейронна мережа (FCNN), що відома як багат шаровий перцептрон (MLP). Мережа, як й багато інших типів, складається із вхідного шару, прихованих шарів та вихідного шару.

Вхідний шар: кількість нейронів відповідає кількості особливостей (фіч) вхідних даних.

Приховані шари: два або більше шарів повнозв'язних нейронів, кількість нейронів у прихованих шарах залежить від складності задачі.

Вихідний шар: один нейрон, оскільки це задача регресії. Вихідний сигнал - прогнозоване числове значення.

Функції активації, що використовуються:

- приховані шари: ReLU (Rectified Linear Unit) або її варіації, як-от Leaky ReLU чи ELU;
- вихідний шар: лінійна активація, оскільки виходом є неперервне числове значення.

Функція втрат визначається виразом (2.3)

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^N L(y^i, f(x^i, \theta)), \quad (2.3)$$

де N - кількість елементів навчальної вибірки;

θ - параметр відображення $f(x)$;

x^i - вектор ознак i -го зразка;

y^i - відповідне i -му зразку значення залежної змінної;

L - втрата функції втрат.

Є багато видів функцій втрат у навчанні з вчителем, наприклад, квадрат евклідової відстані, кросентропія, втрата контрасту (contrast loss), завісні втрати (hinge loss), приріст інформації тощо. Але найчастіше використовується середньо – квадратичне відхилення (RMSE) [8].

Ця нейронна мережа може ефективно вирішувати задачі регресії, аналізуючи складні нелінійні взаємозв'язки в даних. Її гнучкість дозволяє адаптуватися до різноманітних наборів даних і завдань прогнозування.

Широко використовуваним статистичним методом для прогнозування часових рядів є модель ARIMA.

ARIMA - це аббревіатура від Auto-Regressive Integrated Moving Average. Це клас моделей, який фіксує набір різних стандартних тимчасових структур в

даних часових рядів. ARIMA - узагальнення більш простої авторегресійної ковзної середньої і додає поняття інтеграції.

Абревіатура ARIMA носить описовий характер і відображає ключові аспекти самої моделі. Якщо коротко, це:

AR – авторегресія. Модель, яка використовує зв'язок між спостереженням і деякою кількістю більш ранніх спостережень.

I – інтегрована. Використання різниці необроблених спостережень (наприклад, віднімання спостереження з спостереження на попередньому часовому кроці) для того, щоб зробити часовий ряд стаціонарним.

MA - змінна середня. Модель використовує залежність між спостереженням і залишковою помилкою від моделі змінного середнього, застосованої до лагів спостереження.

Кожен з цих компонентів явно вказано в моделі як параметр. Використовується стандартне позначення ARIMA (p, d, q), де параметри замінюються цілочисельними значеннями для швидкої вказівки конкретної використовуваної моделі ARIMA.

Параметри моделі ARIMA визначаються наступним чином:

p - число спостережень відставання, включених в модель, також називається порядком відставання.

d - кількість разів, коли вихідні спостереження розрізняються, також називається ступенем відмінності.

q - розмір вікна ковзної середньої, також називається порядком ковзної середньої.

Побудована модель лінійної регресії, що включає певну кількість і тип термінів, і дані готуються за ступенем різниці, щоб зробити її стаціонарною, тобто видалити трендові і сезонні структури, які негативно впливають на модель регресії.

Значення 0 може бути використано для параметра, який вказує, що цей елемент моделі не використовується. Таким чином, модель ARIMA може бути

налаштована для виконання функції моделі ARMA і навіть простої моделі AR, I або MA.

Прийняття моделі ARIMA для тимчасового ряду передбачає, що базовий процес, який справив спостереження, є процесом ARIMA. Це може здатися очевидним, але допомагає мотивувати необхідність підтвердження припущень моделі в необроблених спостереженнях і в залишкових помилках прогнозів з моделі.

Рекурентна нейронна мережа (RNN) є більш функціональним вибором для прогнозування часових рядів, оскільки вона здатна обробляти послідовності даних та враховувати попередній контекст. Опишемо ключові особливості мережі.

Рекурентна нейронна мережа (RNN) є ефективним інструментом для прогнозування часових рядів, оскільки вона здатна обробляти послідовності даних та враховувати попередній контекст. В RNN використовуються вхідні дані, які можуть бути значеннями часового ряду в певні моменти часу, а також можуть бути інтегровані додаткові фактори, такі як економічні показники або погодні умови, що дає можливість формувати багатовимірний вхідний вектор.

Приховані шари RNN складаються з рекурентних нейронів, які зберігають інформацію з попередніх кроків. Оскільки звичайні RNN часто мають проблеми з довгостроковою пам'яттю, часто використовуються варіанти як LSTM або GRU для кращого збереження контексту.

Вихідний шар зазвичай має один нейрон для прогнозування наступного значення у часовому ряді. Функція активації в цьому шарі залежить від природи даних і може бути лінійною або ReLU.

Тренування RNN відбувається за допомогою модифікованого алгоритму зворотного розповсюдження помилки, званого Backpropagation Through Time (BPTT), та використовує стандартні оптимізатори, як-от Adam або SGD.

Щоб використовувати додаткові фактори у прогнозуванні, важливо інтегрувати ці дані у вхідні дані мережі, а також забезпечити їх нормалізацію для ефективного навчання. Таким чином, RNN можна адаптувати для

прогнозування часових рядів, що залежать не тільки від автокореляції, але й від різноманітних зовнішніх факторів, роблячи її більш гнучкою та ефективною для складних задач прогнозування у порівнянні з моделями ARIMA, які не дозволяють інтегрувати залежності від додаткових факторів.

Після створення програмної реалізації моделі, під час функціонального тестування є доцільним провести порівняльну оцінку якості прогнозування, що дозволяють досягати описані моделей, та виконати вибір однієї з моделей для кожного типу задач проблемного процесора.

2.1.3. Побудова блоків вводу та виводу

Розглянемо тепер побудову блоків вводу та виводу, що мають використовувати мовні моделі.

На сучасному етапі розвитку програмного забезпечення та технологій можна спостерігати явну тенденцію до використання чат-ботів у месенджерах, таких як Telegram, Viber, та інших платформах. Це стає все більш популярним серед користувачів, які надають перевагу зручності та швидкості обміну повідомленнями в уже встановлених додатках, ніж завантаженню нових.

Переваги чат-ботів у месенджерах полягають у їх негайній доступності та зручності використання. Користувачі вже звикли до платформ, які використовують щодня для спілкування з друзями, родиною та колегами. Використання чат-ботів у цих месенджерах забезпечує безшовний досвід, в якому користувачі можуть отримувати інформацію, робити покупки, замовляти послуги або навіть отримувати підтримку без потреби переходити в інший додаток.

Необанки, мережі магазинів, та інші бізнеси, використовують чат-боти для підвищення рівня обслуговування клієнтів. Чат-боти дозволяють швидко відповідати на запити, вирішувати прості проблеми та навіть проводити транзакції, що робить комунікацію з клієнтами більш ефективною та витрато-ефективною.

Ця тенденція також відображає більш широкий тренд у цифровій комунікації - перехід від використання численних додатків до взаємодії через обмежену кількість вже знайомих інтерфейсів. Таким чином, використання чат-ботів у популярних месенджерах можна розглядати як відповідь на постійно зростаючі очікування користувачів щодо зручності та швидкості обслуговування.

Виходячи з наведених міркувань, будуватимемо блоки вводу – виводу саме у вигляді чат – боту у месенджері.

В якості мовних моделей для блоків вводу – виводу системи розглядатимемо GPT-4 від OpenAI та BERT від Google, що пояснюється їх ключовими перевагами:

GPT-4 від OpenAI має широкі можливості для багатозадачності, що дозволяє їй ефективно перекладати запити користувачів на структуровані дані. Вона відома високою точністю у розумінні природної мови і здатністю витягувати ключові факти з вхідних даних. Також, GPT-4 здатна генерувати детальні, зрозумілі відповіді, що робить її ідеальною для перетворення прогнозів та аналізу на природну мову.

BERT від Google відрізняється ефективним обробленням контекстуальних зв'язків у тексті, що є ключовим для точного розуміння запитів користувачів. Ця модель може бути налаштована для специфічних потреб, включаючи аналіз запитів у сфері нерухомості. BERT також ефективно працює з текстами на різних мовах, що може бути важливим для міжнародних запитів.

Ці моделі дозволяють ефективно переводити природну мову у структуровану форму для аналізу проблемним процесором та навпаки, що робить процес прийняття рішень більш інтуїтивним і доступним для користувачів.

Розглянемо на концептуальному рівні шляхи інтеграції мовних моделей, що описані, в програмний засіб. На початку інтеграції необхідно ретельно спланувати, як саме обрані мовні моделі будуть взаємодіяти з іншими

компонентами системи. Це передбачає визначення вхідних і вихідних точок для моделей, способів їх взаємодії з базами даних, користувацьким інтерфейсом та іншими системами.

Моделі як GPT-4 та BERT можуть бути інтегровані як окремі модулі, що працюють у фоновому режимі, або як вбудовані компоненти, що безпосередньо взаємодіють з користувачем через графічний інтерфейс. Оскільки мова йде про використання сторонніх платформ для побудови чат – боту, то інтеграція мовних моделей буде реалізована у вигляді окремих мікросервісів системи, що отримують та передають інформацію до чат – боту з однієї сторони, та до проблемного процесору – з іншої.

2.1.4. Вибір методології розробки та управління проектом

Оскільки розробка використовує певну наукову новизну та нестандартні підходи до відомих задач, важливим аспектом роботи над програмним засобом є перевірка концепцій та методології, що використовується при розробці. Не можна виключати, що перші ітерації функціонального тестування вимагатимуть коригування концепцій або зміни моделей. Тому доцільною є використання методології, що передбачає створення мінімально життєздатного продукту та поступове збільшення його функціональності. Такий підхід в розробці називають «лінійно – ітераційним», він є базою багатьох гнучких (Agile) методологій розробки програмного забезпечення.

Методологія, яка включає створення мінімально життєздатного продукту (MVP) та поступове збільшення функціональності, зосереджена на початковому запуску продукту з базовим функціоналом, достатнім для задоволення первинних потреб користувачів. Цей підхід дозволяє швидко вийти на ринок та отримати зворотний зв'язок від реальних користувачів. Розвиток продукту відбувається через ітерації, кожна з яких включає планування, проектування, розробку, тестування та оцінку, дозволяючи поступово додавати нові функції та вдосконалювати продукт. Цей підхід характеризується гнучкістю та залученням користувачів, що дозволяє швидко

адаптуватися до змін у потребах ринку та зворотного зв'язку. Ітераційний розвиток також сприяє зниженню ризиків, оскільки потенційні проблеми можуть бути виявлені та вирішені на ранніх етапах. Основна мета цього підходу полягає у наданні цінності для користувачів, забезпечуючи регулярне оновлення продукту з новими та вдосконаленими функціями. Цей підхід ідеально підходить для проектів, де потрібна швидка адаптація до змінних умов ринку та потреб користувачів, при цьому забезпечуючи високу якість і надійність кінцевого продукту.

2.2. Загальносистемні рішення

2.2.1. Архітектура системи

Опрацювання загальносистемних рішень включає вивчення сценаріїв використання системи та формулювання функціональних та нефункціональних вимог до програмного забезпечення, вибір архітектурного стилю та патернів, а також планування процесу розробки. Також в рамках розробки загальносистемних рішень створимо контексту діаграму.

В якості архітектури системи будемо розглядати мікросервісну архітектуру, що для розроблюваного програмного засобу має низку переваг над монолітним конструктом через використання низки модулів різної функціональності, функціонування яких є незалежним за виключенням передачі результатів роботи між ними: мовні моделі не застосовуються в проблемному модулі, а хід діалогу з користувачем в чат боті не впливає на аналітичні розрахунки.

Мікросервісна архітектура - підхід у розробці програмного забезпечення, який полягає у створенні незалежних модулів, кожен з яких виконує певну функцію. Ці модулі, або мікросервіси, можуть розроблятися, розгортатися та масштабуватися незалежно один від одного. Основна ідея мікросервісної архітектури полягає в тому, що велике програмне забезпечення може бути розділене на невеликі, легко керовані частини. Кожен мікросервіс

фокусується на виконанні однієї конкретної задачі та має свій власний контекст та базу даних.

Однією з основних переваг мікросервісної архітектури є гнучкість у розробці та підтримці. Оскільки кожен мікросервіс можна розробляти та оновлювати незалежно, це дозволяє командам швидше реагувати на зміни та інновації. Також мікросервісна архітектура забезпечує високу масштабованість, оскільки кожний сервіс можна масштабувати окремо, відповідно до потреб. Ця архітектура також покращує надійність системи, оскільки помилка в одному мікросервісі зазвичай не впливає на роботу інших.

Мікросервіси спілкуються один з одним через легкі протоколи, такі як HTTP/REST або легковагові системи обміну повідомленнями. Це спрощує інтеграцію та взаємодію між різними частинами системи. Використання API дозволяє створювати стандартизовані точки доступу для кожного мікросервісу, що сприяє їх незалежності та взаємодії. Додатково, мікросервісна архітектура сприяє використанню контейнеризації, такої як Docker, що полегшує розгортання та управління сервісами.

Всі ці властивості роблять мікросервісну архітектуру привабливою для сучасних бізнес-додатків, де потрібна швидка розробка, гнучке управління, надійність та масштабованість.

Виходячі з цього, програмний засіб може бути побудований з використанням мікросервісної архітектури, що включає наступні компоненти:

- мікросервіс «Чат – бот»;
- мікросервіс «Інтеграція мовних моделей»;
- мікросервіс «Проблемний процесор»;
- мікросервіс «Робота з БД»
- мікросервіс «Збір та агрегація інформації»

Кожен модуль має відокремлений набір функцій, що також є слідуванням концепції інкапсуляції, та стандартизований контракт взаємодії з

іншими сервісами програмного засобу. Розгортання сервісів доцільно з використанням контейнеризації з подальшою оркестрацією.

2.2.2. Контекстна діаграма системи

Контекстна діаграма - це інструмент візуалізації, який використовується в системному аналізі та розробці програмного забезпечення для виявлення взаємодії системи із зовнішнім світом. Вона показує систему як єдине ціле і контекст, у якому вона функціонує, включно із зовнішніми сутностями (тиками як користувачі, зовнішні системи або процеси), які взаємодіють із нею. Контекстна діаграма системи, що розробляється, показана на рис. 2.1.

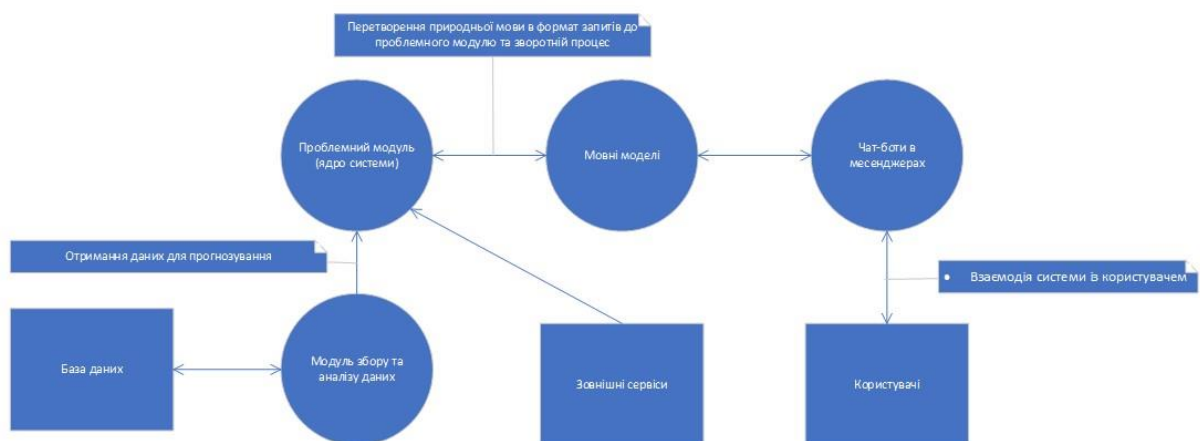


Рисунок 2.1 – Контекстна діаграма системи

Контекстна діаграма містить проблемний модуль та модуль збору даних, що безпосередньо відповідають модулям системи підтримки прийняття рішень; модулі мовних моделей та чат-боту, що реалізують поєднаний блок вводу – виводу СППР. Також на контекстній діаграмі відтворені користувачі системи та джерела інформації: основне – база даних системи, допоміжне – зовнішні сервіси.

2.2.3. Аналіз сценаріїв використання

Розглянемо основні сценарії використання СППР, що розробляється. На рис. 2.2 представлений сценарій розрахунку справедливої вартості об'єктів нерухомості, що знаходяться в портфелі інвестора.

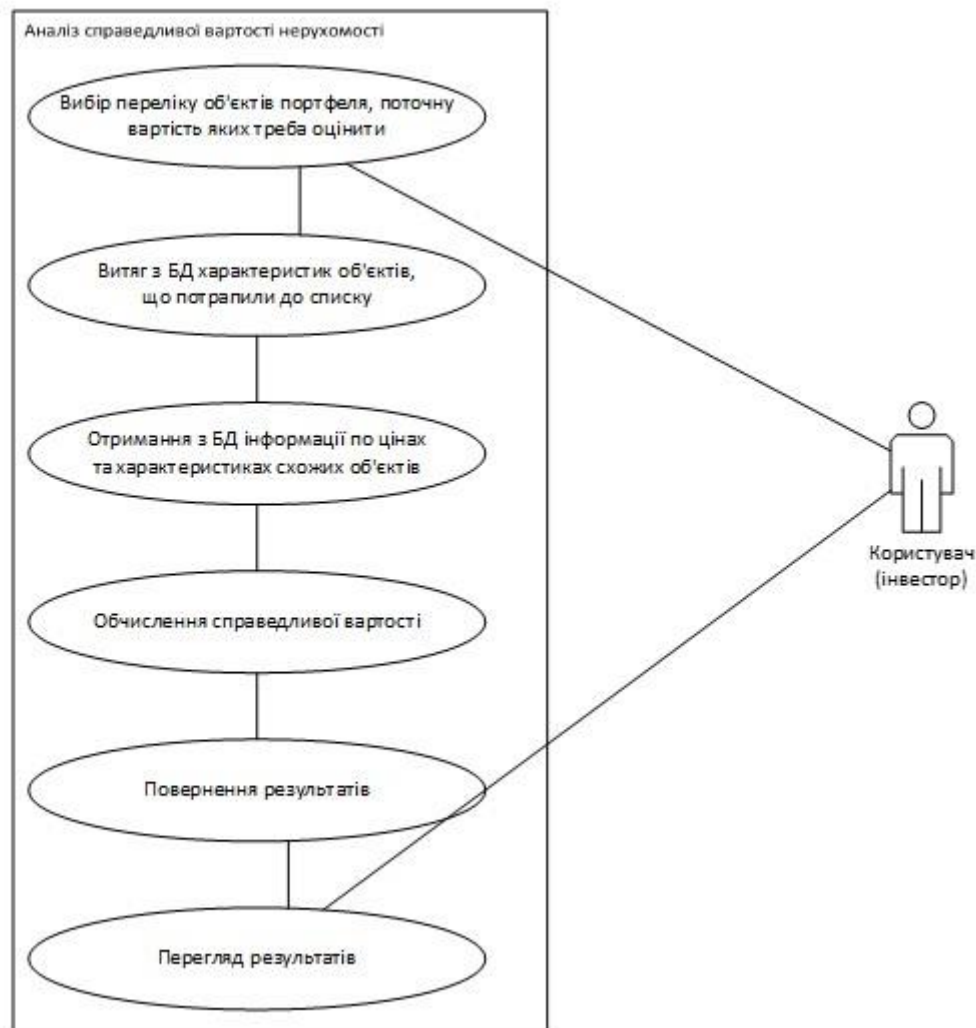


Рисунок 2.2 – Сценарій використання «Аналіз справедливої вартості»

Другим з основних сценаріїв використання є сценарій «Аналіз та прогнозування динаміки вартості», що показаний на рис. 2.3.

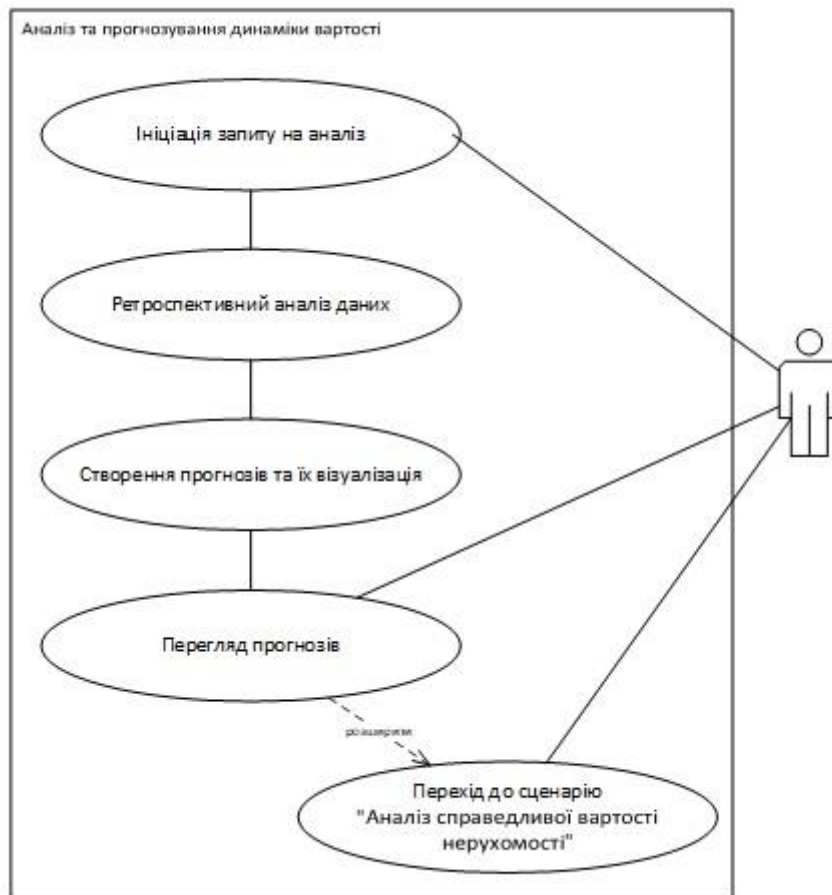


Рисунок 2.3 – Сценарій використання «Аналіз та прогнозування динаміки вартості»

Також до основних сценаріїв відноситься простіший, але частіше використовуваний сценарій «Перегляд вартості оренди», в рамках якого інвестор чи менеджер інвестора може переглядати поточну вартість оренди об'єктів нерухомості та їх частин, що знаходяться в управлінні, та порівнювати цю вартість із середньо ринковими показниками. З сценарію можливий перехід до ініціації зміни вартості оренди або прогнозування динаміки вартості, що дозволить приймати виважене рішення, щодо необхідності зміни вартості оренди в короткостроковій або середньостроковій перспективі.

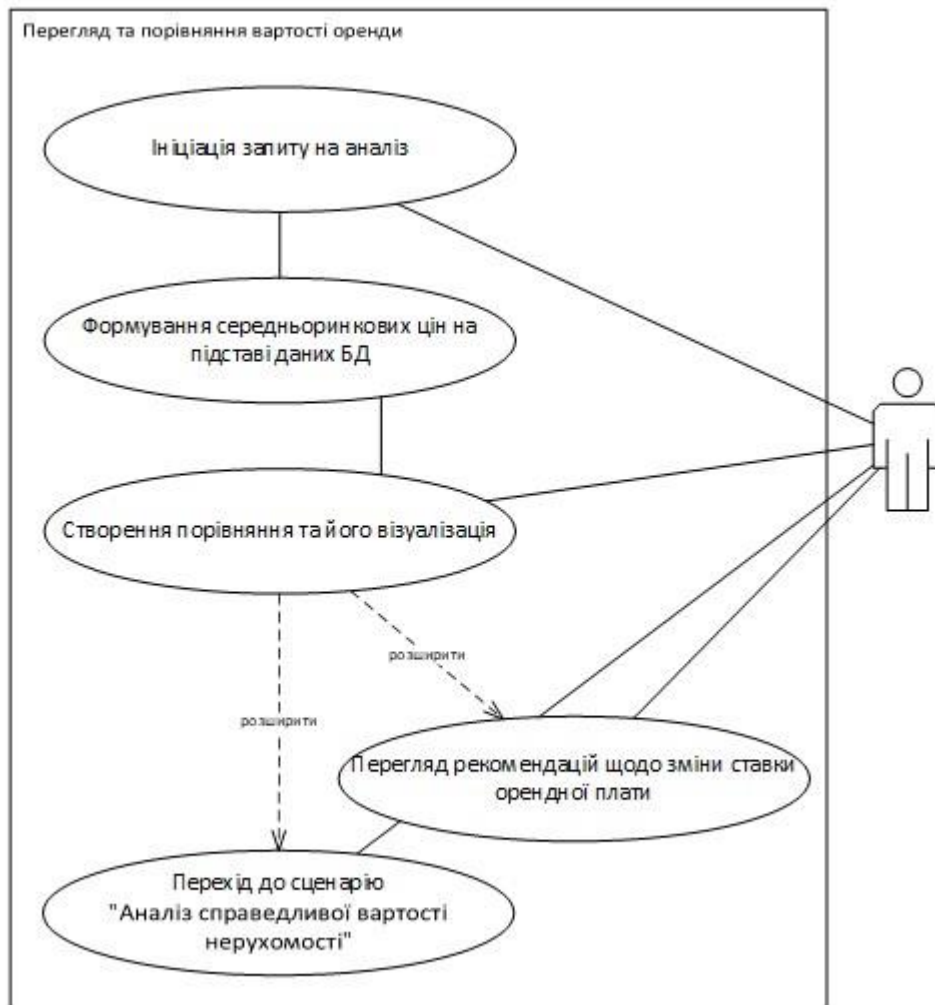


Рисунок 2.4 – Сценарій «Перегляд й порівняння вартості оренди»

2.2.4. Функціональні та нефункціональні вимоги до ПЗ

На підставі проведеного аналізу сценаріїв використання та підсумовуючі попередні підрозділи, можна сформулювати вимоги до системи наступним чином.

Функціональні вимоги.

1. Інтерфейс чат-бота: інтеграція з чат-ботом у месенджерах для взаємодії з користувачем.

2. Перетворення природної мови: використання мовних моделей для перетворення запитів користувачів на структуровані задачі та навпаки, підтримка природнього діалогу з користувачем для уточнення завдань.

3. Аналіз вартості нерухомості: проблемний процесор (модуль) має визначати справедливу вартість об'єктів нерухомості на підставі їх характеристик, що відтворює аналіз ринку в моменті та пропонувати до купівлі об'єкти, що мають нижчу відносно ринкової вартість, але порівнянні характеристики.

4. Аналіз та прогнозування динаміки вартості нерухомості та оренди нерухомості: виконання ретроспективного аналізу вартості нерухомості в інвестиційному портфелі інвестора та побудова прогнозу вартості цієї нерухомості. У випадку позитивного прогнозу – надання рекомендацій щодо розширення портфелю за рахунок подібних об'єктів, у випадку негативного прогнозу – надання рекомендацій щодо реалізації існуючих в портфелі одиниць подібного профілю.

5. Перегляд вартості оренди об'єктів, що входять до портфелю: проблемний процесор виконує порівняльний аналіз вартості оренди об'єктів нерухомості, що входять до портфелю, та їх частин з поточними цінами на ринку нерухомості та прогнозує динаміку. У випадку позитивного прогнозу рекомендує здійснити перегляд цін на наступний період, наприклад, календарний рік, у випадку негативного прогнозу – рекомендує знизити ціни чи надати додаткові послуги чи привілеї орендарям з метою уникнення відтоку орендарів.

6. Обробка та аналіз даних: виконання багатопотокового збору даних, перетворення текстової інформації на цифрово-аналітичну з використанням мовних моделей.

Нефункціональні вимоги.

1. Надійність та стабільність - висока стабільність та надійність роботи системи.

2. Легкість розгортання та масштабованість. Забезпечити легкість розгортання системи в різних середовищах та здатність системи масштабуватися відповідно до зміни обсягів даних та потреб користувачів.

3. Безпека даних - забезпечення безпеки даних та транзакцій в системі.

4. Швидкість та продуктивність - ефективна обробка запитів та аналізу даних з мінімальним затримками.

5. Обробка та аналіз даних - використання Spark, Spark Streaming та Hadoop для збору та аналізу даних.

6. Інтеграція з іншими системами - інтеграція системи з зовнішніми джерелами даних.

Ці вимоги визначають загальний функціонал та ключові характеристики системи, забезпечуючи її ефективне функціонування та задоволення потреб користувачів.

2.3. Рішення з інформаційного забезпечення

До цього етапу розробки включаємо рішення, що пов'язані з управлінням, зберіганням та обробкою інформації в системі.

Планування управління, зберіганням та обробкою інформації в системі будується на базі діаграми сутностей (ER-діаграма), яка дозволяє відтворити структури зберігання інформації в системі. У підрозділі «Рішення з програмного забезпечення» також буде додано окрім загальної діаграми розгортання системи діаграму розгортання кластеру збору та аналізу даних, оскільки він має доволі складну структуру, що жодним чином не відображається на ER-діаграмі у силу різної спрямованості та рівней абстракції цих діаграм.

На ER-діаграмі, що розглядається, показані основні сутності системи й не відтворюються допоміжні, що допомагає зосередитись на суттєвому та не захащувати діаграму. Основними сутностями системи є три типи об'єктів нерухомості. Перший тип «Properties» відтворює об'єкти, що знаходяться у портфоліо інвестора, другий тип «Atomic objects» відтворює частини цих об'єктів, атомарні одиниці на які підписуються договори оренди. Така побудова враховує, що у власності знаходяться цілі об'єкти, а здаватись вони можуть частинами. Якщо об'єкт здається цілком, то в системі все одно

створюються екземпляри двох сутностей, але це може відбуватися без необхідності дублювання при вводі від користувача.

Третій тип сутностей, що описують об'єкти нерухомості, є «Real estate objects», які відповідають іншим об'єктам нерухомості за якими збирається статистика. Тут немає поділу на атомарні об'єкти, оскільки відбувається збирання інформації з відкритих джерел, тобто можна отримати лише характеристики того об'єкту, що здається, в незалежності від того є він частиною чогось більшого або ні.



Рисунок 2.5 – Діаграма сутностей

Зберігання властивостей нерухомості у вигляді сутностей «Property description» з одного боку дозволяє уніфікувати дані про об'єкти, а з іншого – робить гнучким набір характеристик, які потім можуть біти використані при

обчисленнях та візуалізації. Зберігання інформації в такий спосіб дозволяє уникати «хард кодингу» окремих властивостей об'єктів нерухомості, вартості оренди, метражу, наявності комунікацій, тощо. Множина параметрів «Description values» зберігатиметься у окремій таблиці, що не відображається на діаграмі відповідно до принципу, що був позначений вище.

Також важливою характеристикою об'єктів, що винесена в окрему сутність, є їх розташування. Від нього суттєво залежить вартість продажу та оренди, отже, цей параметр має бути добре структурованим та описаним – він гратиме одну з ключових ролей в аналізі та прогнозуванні. Окрім безпосередньо адреси об'єкту використовуватимемо експертну оцінку локації, що втілюється у рейтингу 1-20. На початкових етапах рейтинг проставлятиметься на підставі експертної думки, а накопичивши певну статистику за рейтингами можна запропонувати регресійну оцінку рейтингу за наявними даними про об'єкти.

Безумовно в схемі присутня також таблиця, яка дозволяє фіксувати та відстежувати вартість оренди. На відміну від вартості купівлі – продажу, що зберігається разом з іншими властивостями об'єкту. Така схема зберігання обумовлена тим, що накопичення історії вартості оренди відбувається в суттєво більших об'ємах у порівнянні із статистикою повної вартості.

Для інших об'єктів нерухомості властивості та ціна фіксуються на певну дату, що відображається в полі «snap_date». Це пояснюється тим, що немає стовідсоткової гарантії, що алгоритм зможе регулярно збирати дані про об'єкт з відкритих джерел, тому дані фіксуються на певну дату й виходячи з цієї дати приймають чи не приймають участь в ретроспективному аналізі та прогнозах вартості.

Сутність «User» введено для верифікації входу до системи та фіксації дій користувачів по зміні даних, що зберігаються.

2.4. Рішення з програмного забезпечення

Цей етап ще може бути названий етапом конструювання системи. Він характеризується переходом від абстракції, що використовуються під час опрацювання загальносистемних рішень та моделювання системи, до оперування конкретними елементами програмного комплексу, що розробляється. Під час конструювання програмного забезпечення створюємо діаграму компонентів системи та діаграму розгортання.

2.4.1. Розробка діаграми компонентів

Діаграма компонентів системи представлена на рис. 2.6. Вона відповідає структурним компонентам системи, що були розглянуті під час огляду побудови СППР та контекстній діаграмі, що була представлена раніше у розділі.

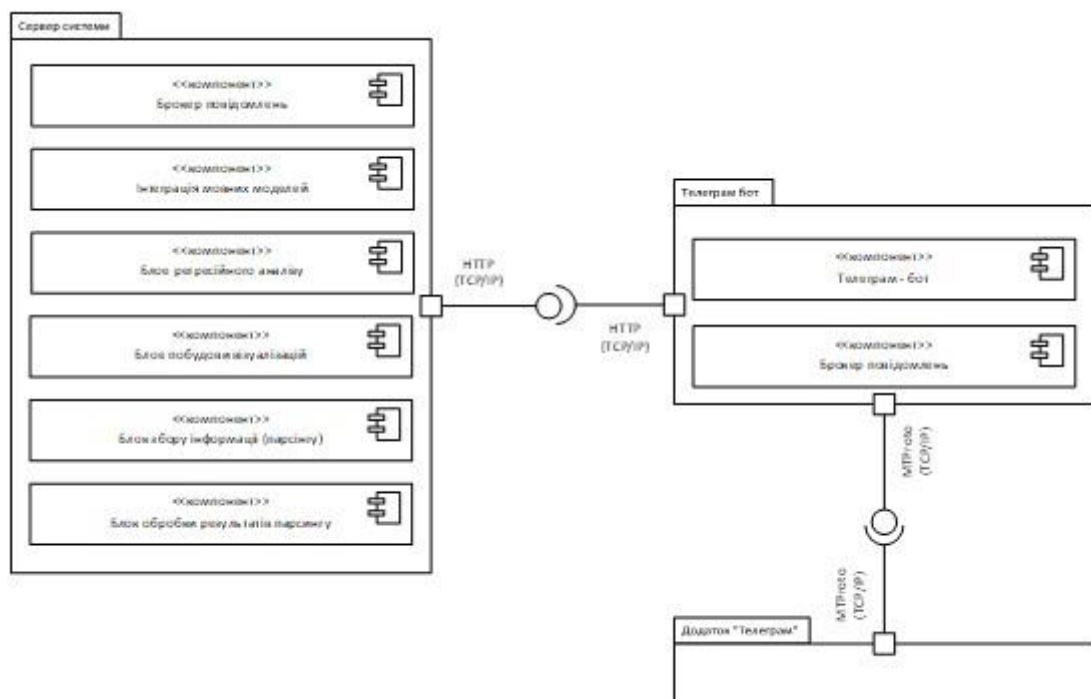


Рисунок 2.6 – Діаграма компонентів.

В якості інтерфейсу користувача виступає додаток «Telegram», що використовує власний протокол MTProto (Mobile Telegram Protocol). Цей

протокол був розроблений командою Telegram і оптимізований для використання в мобільних середовищах. Він забезпечує високу швидкість і безпеку передачі даних. Окремо виділені серверна частина телеграм боту та сервер системи, що «спілкуються» з використанням брокеру повідомлень. Це дозволяє організувати асинхронну роботу системи. Після отримання повідомлення в телеграм бот система надсилає повідомлення (Apache Kafka, Rabbit MQ) до головного серверного додатку систему, що виконує операції з перетворення природньої мови, аналізу, генерації рекомендацій, тощо. Після закінчення обробки система надсилає до серверної частини телеграм боту повідомлення – відповідь й бот вже надсилає його в чат з користувачем. Асинхронність дозволяє одночасно приймати нові запити та обробляти поточні з черги повідомлень. Також подібна побудова системи корисна з точки зору безпеки шляхом ізоляції процесів та інтерфейсів компонентів системи: телеграм бот не звертається безпосередньо до ядра системи.

2.4.2. Розробка діаграми розгортання

Діаграма розгортання системи представлена на рис. 2.7. Вона відтворює архітектуру системи з точки зору фізичних компонентів. Діаграма передбачає, що компоненти розгортаються на одному сервері в окремих контейнерах, що оркеструються з використанням Docker та використовують внутрішню мережу Docker для комунікації.

У разі необхідності масштабування системи можливо розгортання контейнерів на окремих серверах, що дозволяє масштабування тільки тих компонентів, що необхідно. Використання брокерів повідомлень, у тому числі, дозволяє створювати декілька кластерів «проблемного модуля» та управляти надсиланням повідомлень до різних серверів кластеру. Наприклад, до того, що має найменше завантаження, найкоротшу чергу, тощо.

Аналогічним чином можливе розширення кластеру збору, зберігання та аналізу даних, оскільки технологія Apache Spark є спроектованою саме для безперешкодного масштабування у відповідності до потреб.

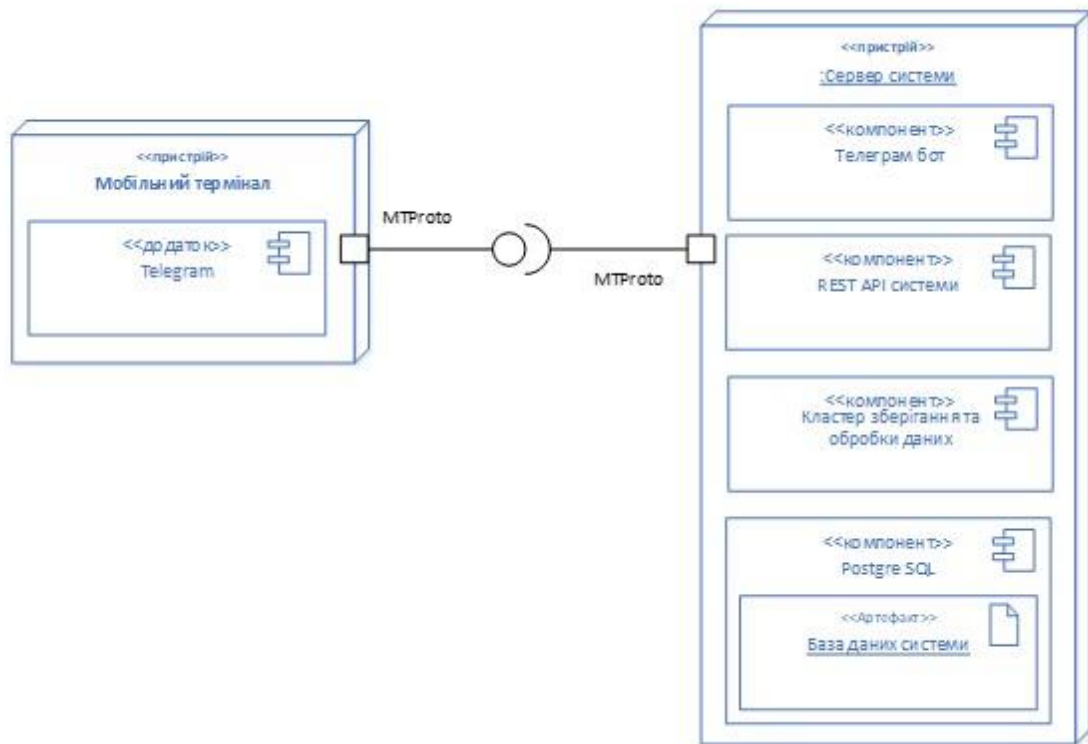


Рисунок 2.7 – Діаграма розгортання системи

Окремо розглянемо побудову кластеру зберігання та аналізу даних. Для потокового збору та впорядкування даних, аналізу існуючих датасетів використаємо кластер даних Spark із розширенням Spark Streaming та Hadoop, для аналізу даних та навчання моделей – хмарні сервіси Google, де за технічної можливості використано графічні ядра для обчислення (Tensorflow орієнтований на використання GPU у той самий час, коли Scikit-learn та XGBoost орієнтовані на використання центрального процесору).

Створена система збору даних, що є необхідною частиною СППР включає в себе:

1. Scrapy павуки для парсінгу даних.
2. Apache Kafka для збору інформації від парсерів.
3. Spark streaming для агрегації та фільтрації даних, що збираються.
4. Hadoop Distributed File System (HDFS) для зберігання зібраних даних.

5. Apache Spark для аналізу зібраних даних та підготовки даних для навчання математичних моделей та штучних нейронних мереж.
6. Apache Zookeeper для координації та синхронізації компонентів кластеру даних.

Графічно схема кластеру роботи з даним представлена на рис. 2.8.

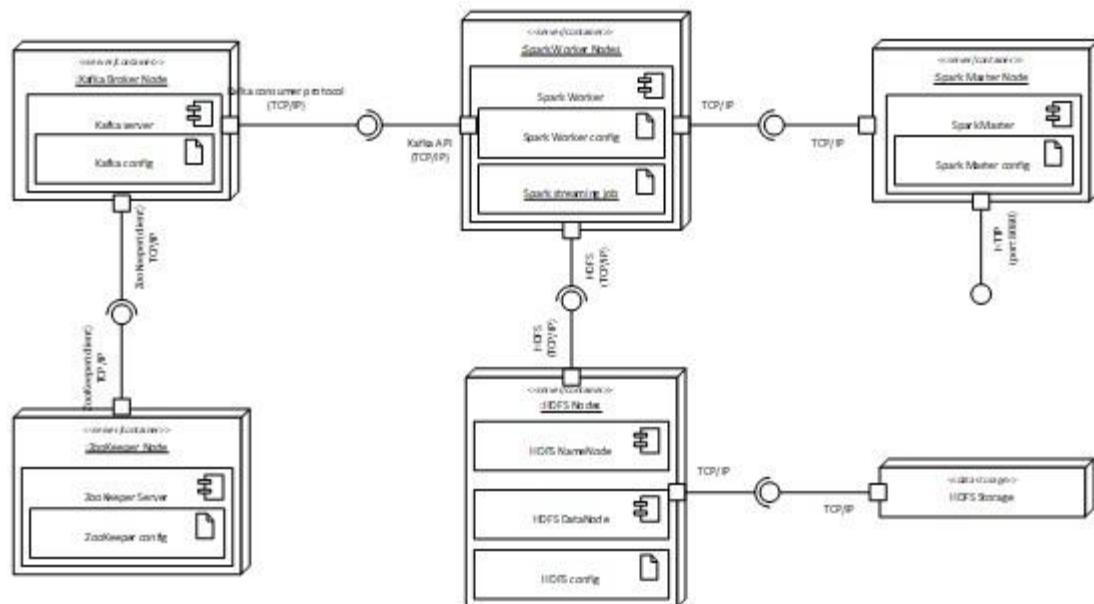


Рисунок 2.8 – Кластер збору даних

Висновки до розділу 2

У другому розділі роботи детально розглянуто ключові аспекти розробки та інтеграції системи підтримки прийняття рішень для управління нерухомістю – від розробки концепції системи до її фізичної архітектури. Це дозволяє спростити процес програмної реалізації та максимально наблизитись до функціональних вимог без додаткових ітерацій циклу розробки.

Основну увагу приділено створенню «проблемного процесору», що є ядром системи та блокам вводу – виводу, що інтегрують мовні моделі та працюють з користувачем через телеграм бот. В рамках розробки концепції СППР охоплено вибір та обґрунтування основних регресійних та мовних моделей, що використовуються.

Функціональні та нефункціональні вимоги до системи визначено на базі контекстної діаграми системи та вивчення сценарії її використання. Формулювання функціональних вимог охоплює різноманітні аспекти, включаючи інтеграцію з чат-ботом, перетворення природної мови, аналіз вартості нерухомості, а також аналіз та прогнозування динаміки вартості та оренди нерухомості. Нефункціональні вимоги зосереджуються на надійності, легкості розгортання, безпеці даних, швидкості та продуктивності системи.

В архітектурному аспекті вирішено використовувати мікросервісну архітектуру, що дозволяє кожному компоненту системи, такому як чат-бот, мікросервіс інтеграції мовних моделей, проблемний процесор та інші, працювати незалежно. Це сприяє гнучкості, масштабованості та надійності системи.

Важливим елементом розробки є планування управління зберіганням та обробкою інформації. Це включає розробку ER-діаграми для відображення структур зберігання інформації, що відображає основні сутності системи та важливі характеристики об'єктів нерухомості, такі як розташування, вартість оренди та інші параметри.

У цілому, другий розділ роботи демонструє комплексний підхід до розробки системи, який охоплює вибір технологій, архітектурні рішення, детальне планування функціоналу та структури даних.

3. РЕАЛІЗАЦІЯ ПРОЕКТУ СППР ДЛЯ УПРАВЛІННЯ ТА ПІДБОРУ ОБ'ЄКТІВ НЕРУХОМОСТІ

3.1. Вибір інструментів реалізації та розгортання СППР

Відповідно до другого розділу система матиме мікросервісну архітектуру, відповідно кожен з сервісів представлятиме собою певну API, що має встановлений контракт взаємодії з іншими мікросервісами в системі. Звісно з тими, з якими має відбуватись обмін повідомленнями, даними, тощо. Окрім брокерів повідомлень інші мікросервіси системи будуватимемо з використанням Python та фреймворку FastAPI.

Python є популярним вибором для розробки REST API за рахунок:

- простоти та читабельності коду, читабельність коду спрощує спільну роботу над проектами і забезпечує більш швидку розробку і підтримку.
- великої спільноти та екосистеми, Python має велику спільноту розробників і широку екосистему бібліотек і фреймворків, що полегшує розробку API. Існує безліч інструментів для створення API, таких як Flask, Django, FastAPI, та інші, які прискорюють розробку і надають готові рішення для обробки HTTP-запитів.
- достатньої продуктивності, яка забезпечується за рахунок асинхронних фреймворків, наприклад FastAPI.
- підтримки сторонніх бібліотек та інструментів, для розробників доступна велика кількість різноманітних бібліотек, які легко інтегруються в продукт, що створюється. Це дозволяє сфокусуватись на бізнес – логіці, а не роботі з рутинними операціями.
- крос-платформеності, Python підтримується на різних операційних системах, це дає змогу розгорнути API на різних платформах.
- достатнього рівня забезпечення безпеки. В репозиторіях Python можна знайти бібліотеки для забезпечення безпеки API, як-от бібліотеки для

автентифікації та авторизації користувачів, оброблення вхідних даних, захисту від атак, як-от SQL-ін'єкції, і багато іншого.

- розширюваності та масштабованості, розроблені API можна легко масштабувати залежно від зростання навантаження і вимог.

Для створення API використовується фреймворк FastAPI [14]. Він розроблений з акцентом на швидкість, продуктивність і автоматичну генерацію інтерактивної документації, що робить його чудовим вибором для розробки високопродуктивних і документованих API. Ось деякі ключові особливості та переваги FastAPI:

- швидке розгортання FastAPI дає змогу швидко створювати API завдяки інтуїтивному синтаксису й автоматичній генерації маршрутів на основі анотацій Python.
- FastAPI повністю підтримує асинхронне програмування, що дає змогу обробляти безліч одночасних запитів без блокування, збільшуючи продуктивність.
- автоматична валідація запитів FastAPI, яка полягає в тому, що автоматично виконується валідація вхідних запитів на основі анотацій Python, це зменшує ймовірність помилок і спрощує обробку невірних запитів.
- FastAPI автоматично генерує інтерактивну документацію, засновану на анотаціях Python і коментарях до коду. Це робить процес документування API простішим та більш наочним.
- FastAPI підтримує інтеграцію з популярними базами даних, такими як PostgreSQL, MySQL, SQLite та іншими, що забезпечує гнучкість у роботі з даними.
- FastAPI створений з урахуванням оптимізації продуктивності, і він спроектований так, щоб забезпечувати високу швидкість обробки запитів.

Модуль, що використовує математичні моделі та машинне навчання також побудовано на Python.

В якості бази даних, як вже було зазначено раніше, використовується PostgreSQL. PostgreSQL – це потужна, відкрита і багатофункціональна реляційна база даних, яка підтримує як SQL, так і JSON для нереляційних запитів. Її вибір базується на низці переваг, що роблять цю СУБД найкращим вибором для додатків різноманітних типів за рахунок:

- високої надійності, підтримки транзакцій, відкату та журналізації, що забезпечує безпеку даних;
- ефективної обробки великих обсягів даних, що є важливим для додатків управління складом, де може бути велика кількість транзакцій;
- підтримці складних SQL запитів та об'єднанню даних, яке дозволяє створювати гнучкі та потужні запити для аналізу та управління запасами;
- широкої підтримки розширень, що збільшують можливості СУБД.

Для роботи з віддаленою базою даних використано фреймворк (ORM) SQLAlchemy версії 2.0. SQLAlchemy забезпечує абстракцію від конкретної СУБД, що дає змогу писати код, який працюватиме з різними базами даних, не змінюючи його структуру.

Однією з основних переваг використання ORM, таких як SQLAlchemy, є абстракція, яка відокремлює бізнес-логіку додатку від конкретних деталей бази даних. Це означає, що розробники можуть взаємодіяти з базою даних, використовуючи високорівневі об'єктні конструкції, не турбуючись про SQL синтаксис. Такий підхід дозволяє легко змінювати тип СУБД (наприклад, з MySQL на PostgreSQL) без необхідності переписування запитів або структури коду.

Використання ORM спрощує процес розробки, оскільки розробники можуть використовувати об'єктно-орієнтований синтаксис для взаємодії з базою даних. Це зменшує кількість коду, необхідного для виконання базових

операцій з базою даних, та дозволяє розробникам концентруватися на більш важливих аспектах додатку. Окрім того, ORM, як правило, включає в себе різноманітні засоби для оптимізації роботи з базами даних, такі як кешування запитів та лейзі-лоадінг (відкладене завантаження).

Використання ORM допомагає забезпечити більш високий рівень безпеки, особливо щодо SQL ін'єкцій, оскільки запити генеруються автоматично та часто очищаються від потенційно шкідливих вхідних даних. Також, використання об'єктно-орієнтованого підходу може допомогти уникнути поширених помилок, які виникають при ручному написанні SQL запитів, та забезпечує більш інтуїтивно зрозумілу структуру коду, що сприяє легшому виявленню та виправленню помилок.

Тепер розглянемо компоненти, які використовуються при потоковому зборі та обробці даних, оскільки в БД зберігаються вже очищені та переформатовані належним чином дані.

Spark - інструмент для обробки даних, що працює на принципах розподіленої обробки. Він дозволяє обробляти великі обсяги даних швидше, ніж традиційні підходи завдяки своїй здатності обробляти дані в пам'яті (in-memory processing). Spark підтримує мови програмування, такі як Scala, Python та Java, і забезпечує багатий набір високорівневих інструментів, включаючи Spark SQL для запитів до даних, MLlib для машинного навчання, GraphX для обробки графів і Spark Streaming.

Hadoop - фреймворк для зберігання та обробки великих обсягів даних, розподілених по кластерах. Основними компонентами Hadoop є Hadoop Distributed File System (HDFS) для зберігання даних та MapReduce для обробки даних. Hadoop відомий своєю масштабованістю та надійністю, оскільки він може зберігати та обробляти величезні обсяги даних.

Spark Streaming є компонентом Apache Spark, який дозволяє обробляти поточкові дані. Він може отримувати дані в реальному часі з різних джерел, таких як Kafka, Flume, Twitter, ZeroMQ чи TCP сокети, і дозволяє використовувати ті ж самі API, які використовуються для обробки пакетних

даних. Spark Streaming ділить вхідний потік даних на маленькі пакети, які потім обробляються Spark. Це дозволяє проводити складні обчислення в реальному часі і повертати результати майже миттєво. Він також інтегрується з Hadoop та іншими фреймворками обробки даних, що робить його гнучким рішенням для потокової обробки даних у великих масштабах.

На завершення інструментів реалізації та розгортання системи розглянемо контейнеризацію та оркестрацію Docker.

Використання контейнеризації Docker має кілька значних переваг, як для тестування на локальній машині, так й у разі промислової експлуатації на віддаленому сервері чи на двох серверах. Перше та найголовніше, це підвищення надійності, оскільки окремі контейнери для мікросервісів та бази даних забезпечують ізоляцію процесів, що зменшує ризик взаємного впливу помилок або збоїв в одній системі на іншу.

Друга перевага полягає в гнучкості та масштабованості. Використання контейнерів дозволяє легко масштабувати кожен компонент незалежно. Наприклад, можна збільшити кількість контейнерів для обробки запитів REST API під час пікових навантажень, не торкаючись бази даних. Це забезпечує оптимальне використання ресурсів та підтримку високої продуктивності системи.

Третя перевага - це спрощення процесу розгортання та тестування. Розміщуючи REST API та базу даних у окремих контейнерах, можна легко відтворити робоче середовище (production) на локальній машині для тестування. Це забезпечує високу точність тестування, оскільки середовище розробника максимально наближене до середовища комерційної експлуатації (production). Крім того, з допомогою технологій контейнеризації, таких як Docker, процес налаштування середовища стає більш стандартизованим та автоматизованим.

Четверта перевага - підвищення безпеки. Розділення REST API та бази даних на різні контейнери дозволяє більш точно налаштувати мережеві політики та контроль доступу для кожного компоненту. Це знижує ризик

несанкціонованого доступу та можливість атак, оскільки для компрометації всієї системи зловмисникам потрібно проникнути в кілька ізольованих середовищ. Крім того, використання окремих контейнерів полегшує відстеження та аналіз безпекових подій.

Отже, розгортання серверної (віддаленої) частини програмного комплексу виконуємо за допомогою Docker, для збірки контейнерів сервісів – Docker build, для оркестрації контейнерів – Docker compose.

3.2. Розгляд контрольних прикладів

3.3. Методика тестування створеного програмного забезпечення

Unit-тестування мікросервісів, написаних на Python, так само як інших програмних продуктів, є базовим елементом забезпечення якості програмного продукту. Процес тестування включає такі кроки, як:

1. Підготовка середовища тестування, його налаштування середовища для unit-тестування. Це включає створення віртуального середовища Python для ізоляції залежностей і налаштування необхідних інструментів тестування - unittest.
2. Розробка тестових випадків, які перевіряють функціональність окремих компонентів мікросервісу. Це може включати тестування різних HTTP-методів (GET, POST, PUT, DELETE), перевірку відповіді сервера (коди статусу, формати відповідей) та обробку помилок. Для обчислювальних моделей, якість яких вже було попередньо перевірено під час дослідження, формують тести, що перевіряють відповідність результатів розрахунків очікуваням. В цілому метою створення тестових випадків є перевірка відповідності поведінки системи очікуваній в різних операціях, як атомарних, так й більш складних.
3. Мокування зовнішніх залежностей, що ізолюють тести від залежностей, таких як бази даних або зовнішні сервіси. Це можна здійснити за

допомогою мокування (mocking) - техніки створення імітаційних об'єктів, які імітують поведінку справжніх об'єктів. Такий підхід дозволяє досягти ідемпотентності результатів, що мають бути отримані під час тестування.

4. Виконання тестів та аналіз результатів, під час яких власне перевіряють, чи працюють окремі частини API належним чином. У разі виявлення помилок, вони повинні бути виправлені, а тести повторно виконані.

Coverage у контексті unit-тестування відіграє ключову роль, оскільки покриття вказує на відсоток коду, який охоплюється тестами. Високий рівень покриття тестами є важливим, оскільки це забезпечує більшу впевненість у тому, що програмний код працює належним чином і помилки будуть виявлені на ранньому етапі. Максимальне покриття допомагає зменшити ризики у випуску продукту, забезпечує стабільність і надійність програмного забезпечення. Це особливо важливо в складних системах, де непередбачені взаємодії між компонентами можуть призвести до критичних помилок.

Важливим кроком, що передує промисловій експлуатації системи, є інтеграція з системою безперервної інтеграції (CI): Для автоматизації процесу тестування та забезпечення стабільності коду, unit-тести можуть бути інтегровані у систему безперервної інтеграції, яка автоматично виконує тести при кожному оновленні коду.

Тестування проводиться в наступному порядку:

1. Unit-тести для окремих компонентів. Написання unit-тестів для індивідуальних елементів інтерфейсу та логіки обробки даних. Це включає перевірку правильності роботи форм, валідацію введених даних та обробку помилок.
2. Тести мережевих запитів. Основну частину тестування мережевих запитів виконаємо під час тестування REST API з використанням Postman та requests.

3. Інтеграційні тести. Інтеграційне тестування перевіряє взаємодію між різними частинами додатку, включаючи мережеві запити, використання мовних моделей та інтерфейс телеграм – боту.

Після виконання описаних етапів тестування переходять до тестування відповідності розробленого програмного забезпечення функціональним та нефункціональним вимогам, що є ключовим етапом у процесі комерційної розробки ПЗ. Цей процес має на меті забезпечення відповідності розробленого програмного забезпечення всім зазначеним вимогам та очікуванням замовника чи кінцевого користувача.

1. Функціональне тестування. Цей тип тестування зосереджений на перевірці функцій програмного забезпечення. Він включає перевірку конкретних дій чи функцій, які програма має виконувати. Функціональне тестування перевіряє, чи правильно система відповідає на вхідні дані, виконує свої функції, та чи видає відповідні вихідні дані. На цьому етапі тестування використовують стандартний користувацький інтерфейс або імітують запити до нього, не звертаючись безпосередньо до інших компонентів програмного засобу, як це відбувається під час unit або інтеграційних тестів.

2. Нефункціональне тестування. На відміну від функціонального, нефункціональне тестування зосереджене на аспектах програмного забезпечення, які не стосуються конкретних функцій, але є важливими для загальної якості системи. Це включає перевірку продуктивності (як швидко система обробляє запити), надійності (стабільності роботи програми), сумісності (здатності працювати з різними операційними системами, браузерами тощо), масштабованості, витривалості, зручності користування, і доступності.

З особливостей тестування на відповідність вимогам можна виділити:

- тестові сценарії виконуються вручну або за допомогою автоматизованих інструментів. Це включає введення даних, виконання функцій та перевірку відповідності отриманих результатів очікуваним.

- результати тестування документуються, включаючи інформацію про виконані тести, виявлені проблеми та рекомендації щодо їх виправлення.
- тестування може проводитися в кілька ітерацій, особливо після внесення змін або виправлення помилок, щоб переконатися у повній відповідності ПЗ усім вимогам.

Результат тестування на відповідність функціональним та нефункціональним вимогам представлений в наступному підрозділі у вигляді таблиці перевірки.

3.4. Перевірка відповідності програмного засобу вимогам

Результат тестування на відповідність функціональним та нефункціональним вимогам представлений у вигляді таблиці перевірки 3.1.

Таблиця 3.1 - Перевірка відповідності розробки вимогам то ПЗ

№ з/п	Формулювання вимоги	Фактична реалізація
1	Реалізація користувацького інтерфейсу системи у вигляді чат – боту	Інтерфейс системи реалізовано у вигляді чат – боту.
2	Перетворення прямої мови на структуроване завдання та зворотне перетворення результатів роботи «проблемного процесора» з використанням мовних моделей	Для перетворення природньої мови підключено ChatGPT шляхом запитів до Open AI API.

Продовження таблиці 3.1

№ з/п	Формулювання вимоги	Фактична реалізація
3	Аналіз вартості нерухомості «проблемним процесором»	Вартість нерухомості аналізується та порівнюється з поточним станом ринку з використанням регресійних моделей.
4	Аналіз та прогнозування динаміки вартості нерухомості та оренди нерухомості	Аналіз та прогнозування динаміки вартості нерухомості використовуються шляхом накопичення необхідної статистики та застосування статистичних моделей прогнозування часових рядів.
5	Перегляд вартості оренди об'єктів, що входять до портфелю	Користувач може запитати систему про вартість та історію вартості об'єктів, що входять до портфелю.
6	Обробка та аналіз даних	Створено кластер для збору та потокової обробки даних, необхідних для функціонування проблемного процесору системи.
7	Надійність та стабільність	Надійність та стабільність досягається за рахунок використання конфігурації системи, що використовується й в промисловій розробці, та застосування UNIX – подібних систем для розгортання

Продовження таблиці 3.1

№ з/п	Формулювання вимоги	Фактична реалізація
8	Легкість розгортання та масштабованість	Легке розгортання та масштабованість досягаються за рахунок контейнеризації та оркестрації Docker.
9	Безпека даних	Використовані сховища даних мають вбудовані протоколи безпеки, REST API та база даних знаходяться в окремих, ізольованих контейнерах, застосовується шифрування для чутливих даних.
10	Швидкість та продуктивність	Швидкість та продуктивність досягаються за рахунок асинхронності фреймворку та використання UNIX – подібних систем для розгортання.
11	Інтеграція з іншими системами	API «проблемного процесора» дозволяє інтеграцію із сторонніми системами.

Висновки до розділу 3

Розділ підкреслює важливість ретельного вибору технологій та інструментів для обробки та аналізу даних, а також необхідність всебічного тестування для забезпечення якості та відповідності програмного забезпечення вимогам проекту. Також розглядається низка контрольних

прикладів, які дозволяють як перевірити функціонування системи, так й продемонструвати її використання, а також надано результати перевірки відповідності програмного засобу функціональним та нефункціональним вимогам, що були сформульовані у розділі 2.

У розділі акцентується на значущості Python у якості основної мови програмування, а також на використанні FastAPI як ефективного інструменту для створення мікросервісів. Це включає обговорення переваг такого вибору, зокрема з точки зору продуктивності та легкості розробки.

Висвітлено значення PostgreSQL як надійної системи управління базами даних та SQLAlchemy як інструменту для взаємодії з цією СУБД у контексті Python. Використання цих технологій забезпечує ефективність у роботі з даними, спрощує процес розробки та підвищує гнучкість системи.

Розділ деталізує процес тестування розробленого програмного забезпечення, включаючи unit-тестування мікросервісів, використання моків для ізоляції тестів від зовнішніх залежностей, а також оцінку покриття коду тестами. Це демонструє важливість якісного тестування для забезпечення надійності, стабільності та відповідності розробленого ПЗ заданим вимогам.

Разом ці аспекти відображають комплексний підхід до розробки та тестування програмного забезпечення, в якому використання сучасних інструментів та методологій гарантує ефективність, гнучкість та високий рівень якості кінцевого продукту.

4. РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ РОЗРОБКИ СППР

4.1. Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку системи підтримки прийняття рішень у сфері нерухомості. Система базується на сучасній мікросервісній архітектурі, використовує актуальні методи машинного навчання та конфігурації штучних нейронних мереж, а також трендові мовні моделі для спілкування з користувачем природньою мовою. В якості інтерфейсу системи використовується телеграм – бот, що звільняє користувача від необхідності встановлення додаткового програмного забезпечення на мобільний телефон, лептоп або стаціонарний ПК. Особливістю програми є створення інтегрованого рішення, яке враховує широкий спектр факторів. Новизна полягає в створенні СППР класичної компоновки з використанням сучасної архітектури, моделей та використанням мовних моделей в блоках вводу та виводу.

Прямих аналогів у створеній системі немає, оскільки СППР у більшості випадків не передбачають створення типових рішень окрім галузі управлінського обліку на підприємствах, де вони реалізовані як просунутий блок системи управлінського та бухгалтерського обліку.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу здійснюють із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 4.1.

Таблиця 4.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Критерій	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					
2	Багато аналогів на малому ринку	Ринкові п 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 трохи 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
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
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні ресурси на навчання наявних.	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так із комерційної реалізації ідеї

Продовження таблиці 4.1

Практична здійсненність					
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

Усі дані по кожному параметру занесено в таблиці 4.2

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	4	3	3
Наявність аналогів на ринку	4	4	4
Цінова політика	3	4	3
Технічні та споживчі властивості виробу	4	3	3

Продовження таблиці 4.2

Критерії оцінювання	ПІБ експертів	Критерії оцінювання	ПІБ експертів
	Експерт 1		Експерт 1
Експлуатаційні витрати	3	4	3
Ринок збуту	3	4	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	3	4	3
Матеріально-технічна база	3	3	4
Термін реалізації ідеї	4	4	3
Супровідна документбация	4	3	4
Сума	42	42	44
Середньоарифметична сума балів	$(44+43+41) / 3 = 42,66$		

За даними таблиці 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів 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

Як видно з таблиці, рівень комерційного потенціалу розроблюваного програмного продукту є високим, що досягається за рахунок сучасної архітектури та використання актуальних математичних моделей для регресійного аналізу та аналізу часових рядів. Також балів додає використання мовних моделей в блоках вводу – виводу. Особливістю програми є створення інтегрованого рішення, яке враховує широкий спектр факторів, що мають бути враховані при аналізі цін на ринку нерухомості та їх динаміки.

4.2. Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи.

Основна заробітна плата розробників розраховується за формулою

$$З_0 = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p} \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника, грн.;

T_p – число робочих днів в місяці, 22 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 4.4.

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	33199	1509,06	42	63380,52
Програміст	27555	1252,53	42	52606,26
Всього				115986,78

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Додаткова заробітна плата розробників.

Додаткова заробітна плата розробників, які приймали участь в розробці обладнання розраховується як 14,3% від основної заробітної плати розробників та робітників:

$$З_д = \frac{З_0 \cdot 14,3 \%}{100\%} \quad (4.2)$$

$$З_д = (115986,78 \cdot 14,3 \% / 100 \%) = 16\,586,10 \text{ (грн.)}$$

Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = \frac{(З_о + З_д) \cdot 22 \%}{100\%} \quad (4.3)$$

$$Н_з = (115986,78 + 16\,586,10) \cdot 22 \% / 100 \% = 29166 \text{ (грн.)}$$

Витрати на матеріали та комплектуючі.

Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

Амортизація обладнання

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$А = \frac{Ц}{Т_в} \times \frac{t_{\text{вик}}}{12} \text{ (грн.)} \quad (4.4)$$

де Ц – балансова вартість обладнання, грн.;

Т – термін корисного використання обладнання згідно податкового законодавства, років;

$t_{\text{вик}}$ – термін використання під час розробки, місяців.

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 23000 грн., термін його корисного використання згідно податкового законодавства — 2 роки, а термін його фактичного використання — 1,7 міс.

$$A = \frac{23000}{2} \times \frac{1,7}{12} = 1628,4 \text{ (грн.)} \quad (4.5)$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення.

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн. (Visual Studio Professional 1800 грн/міс, використовувався 1,82 місяці), то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $B_{\text{нем.ак.}} = 135040$ грн.

Розрахунки заносимо до таблиці 4.5.

Таблиця 4.5 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія (HP Pavilion)	23000	2	1,7	1628,4
Офісне обладнання (меблі)	20600	4	1,7	757,576
Приміщення	1920000	20	1,7	13600
Всього				15985,96

Тарифи на електроенергію для непобутових споживачів

Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$B_e = B \cdot \Pi \cdot \Phi \cdot K_{\Pi} \quad (4.6)$$

де B – вартість 1 кВт-години електроенергії для 1 класу підприємства,

$B = 6$ грн./кВт;

Π – встановлена потужність обладнання, кВт. $\Pi = 0,3$ кВт;

Φ – фактична кількість годин роботи обладнання, годин;

K_{Π} – коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$B_e = 0,9 \cdot 0,3 \cdot 8 \cdot 42 \cdot 6 = 544,32 \text{ (грн.)}$$

Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників

$$I_B = (3_0 + 3_p) \cdot \frac{H_{IB}}{100\%} \quad (4.7)$$

де H_{IB} — норма нарахування за статтею «Інші витрати».

$$I_B = 115986,78 \times \frac{59\%}{100\%} / = 68432,2 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми

основної заробітної плати дослідників:

$$H_{\text{нзв}} = (3_o + 3_p) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (4.8)$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{\text{нзв}} = 115986,78 \cdot \frac{138\%}{100\%} = 160061,75 \text{ (грн.)}$$

Витрати на проведення науково-дослідної роботи

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$\begin{aligned} B_{\text{заг}} = & 115986,78 + 16\,586,10 + 29166 + 15985,96 + 13540 + 544,3 \\ & + 68432,2 + 160061,75 = 432503,11 \text{ грн.} \end{aligned}$$

Загальні витрати на науково – дослідну діяльність

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються ЗВ, визначається за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\eta} \text{ (грн) \%} \quad (4.9)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$;

розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Обираємо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 432503,11 / 0,5 = 865\,006,22 \text{ грн.}$$

4.3. Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнюювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

- вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;
- зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);
- кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;
- визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

Розробка чи суттєве вдосконалення програмного засобу

Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\P_i = (\pm\Delta\P_0 \cdot N \cdot \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (4.10)$$

де $\pm\Delta\P_0$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_0 = \Pi_b \pm \Delta\P_0$;

Цб – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість і ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$;

p – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 5000 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 1000 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року — на 2000 шт., протягом другого року — на 4000 шт., протягом третього року на 6000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\begin{aligned}\Delta\P_1 &= (0 \times 1000 + (5000 + 1000) \times 2000) \times 0,8333 \times 0,27 \times (1 - 0,18) \\ &= 2,213,911,44 \text{ грн.}\end{aligned}$$

$$\begin{aligned}\Delta\P_2 &= (0 \times 1000 + (5000 + 1000) \times (2000 + 4000) \times 0,8333 \times 0,27) \times \\ &\times (1 - 0,18) = 6641734,32 \text{ грн.}\end{aligned}$$

$$\begin{aligned}\Delta\P_3 &= (0 \times 1000 + (5000 + 1000) \times (2000 + 4000 \\ &+ 6000) \times 0,8333 \times 0,27) \times (1 - 0,18) = 22139114,4 \text{ грн.}\end{aligned}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 22139114,4 грн.

Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Розраховуємо приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.11)$$

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

T – період часу, протягом якого виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\begin{aligned} ПП &= (2,213,911,44/(1 + 0,1)^1) + (6641734,32/(1 + 0,1)^2) \\ &\quad + (1328346,64/(1 + 0,1)^3) = \\ &= 2\,012\,646,76 + 6\,716\,651,60 + 12794402,19 \\ &= 24141914,14 \text{ грн.} \end{aligned}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \times ЗВ, \quad (4.12)$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію і це можуть бути витрати на

підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}}=2...5$, але може бути і більшим;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 \times 865006,22 = 1730012,44 \text{ грн.}$$

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = \text{ПП} - PV, \quad (4.13)$$

$$E_{\text{абс}} = 24141914,14 - 1730012,44 = 22411901,7 \text{ грн.}$$

Оскільки $E_{\text{абс}} > 0$, то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_{\text{в}}$. Для цього використаємо формулу

$$E_{\text{в}} = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (4.14)$$

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

$$E_{\text{в}} = \sqrt[3]{(1 + 22411901,7/1730012,44)} - 1 = 1,407$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (4.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = (0,09...0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,5)$.

$$\tau_{\min} = 0,14 + 0,1 = 0,24.$$

Так як $E_v > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{\text{ок}} = \frac{1}{E_v}, \quad (4.16)$$

$$T_{\text{ок}} = \frac{1}{1,407} = 0,7 \text{ р.}$$

Оскільки $T_{\text{ок}} < 3$ -х років, а саме термін окупності рівний 0,7 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу 4

Розділ 4 присвячений розрахунку економічної ефективності впровадження системи підтримки прийняття рішень (СППР) у сфері нерухомості. Оцінка базується на тому, що розроблено систему на базі мікросервісної архітектури, що включає методи машинного навчання, конфігурації штучних нейронних мереж, а також мовні моделі для спілкування з користувачем. Використання телеграм-бота як інтерфейсу звільняє

користувача від необхідності встановлення додаткового програмного забезпечення.

Система не має прямих аналогів на ринку, відрізняючись використанням сучасних технологій у класичній компоновці СППР. Комерційний та технологічний аудит розробки проводився з допомогою трьох незалежних експертів, які оцінювали розробку за 12 критеріями в п'ятибальній системі оцінювання.

За результатами оцінювання, розробка має високий комерційний потенціал, завдяки інноваційності, відсутності аналогів на великому ринку, конкурентоспроможній ціні, перевагам у технічних та споживчих властивостях, а також оптимізації експлуатаційних витрат. Особливістю програми є її інтегрований підхід, що враховує широкий спектр факторів, включаючи погодні умови, для прогнозування попиту на транспортні послуги.

Розрахунки економічної частини включають оцінку витрат на розробку, визначення чистого прибутку від реалізації продукту, розрахунок періоду окупності та загального економічного ефекту. Встановлено, що загальні витрати на розробку складають 865006 гривень, а розрахунковий чистий прибуток і період окупності інвестицій свідчать про високу економічну ефективність та конкурентоспроможність розробки. Розроблений продукт є дешевшим за аналоги та має високу конкурентоспроможність, з періодом окупності приблизно 0,7 року, що робить його привабливим для інвесторів та виробників.

5. ОХОРОНА ПРАЦІ

5.1. Вступ

Основною метою охорони праці є забезпечення безпеки працівників і запобігання травмам, захворюванням та нещасним випадкам на роботі, що допомагає зберегти життя та здоров'я працівників. Травми та захворювання, пов'язані з небезпечними умовами праці, можуть призвести до високих витрат на медичне лікування, відновлення та компенсації. Забезпечення безпечних та комфортних умов праці сприяє покращенню загального стану здоров'я працівників, що може призвести до підвищення їхньої продуктивності та ефективності.

Аналіз нещасних випадків на виробництві показує, що основними їх причинами в Україні є організаційні (до 60%), а це, перш за все, зумовлено низьким рівнем управління охороною праці, відсутністю необхідних знань з охорони праці на всіх ланках виробництва, умінь та навичок безпечної праці, належного контролю за станом виробничого середовища та недоліками правового регулювання трудових відносин.

Безпека людини нерозривно пов'язана з оточуючим її виробничим середовищем. Останнє характеризується породжуваними діяльністю людини об'єктами, явищами, фізичними, хімічними, біологічними та соціальними факторами, які прямо чи опосередковано впливають на самопочуття та стан здоров'я працюючих.

Структурна схема охорони праці тісно пов'язана з рядом інших дисциплін: безпекою життєдіяльності, основами охорони праці, правознавством, схематично це наведено на рис.5.1.



Рисунок 5.1 - Структурна схема охорони праці

Правові та організаційні основи охорони праці є тією базою, яка забезпечує соціальний захист працівників і на якій будується інженерно-технічна складова охорони праці. Виробнича санітарія, виробнича безпека та пожежна безпека на виробництві, з одного боку, базуються на правових та організаційних основах охорони праці, з іншого боку, вони визначають пріоритети, структуру цих основ та необхідність змін у них. Виробнича санітарія, виробнича безпека та пожежна безпека на виробництві також тісно пов'язані між собою.

Таким чином, завдання охорони праці – звести до мінімуму ймовірність травматизму чи захворювання працюючих та створити оптимальні умови для їх праці, що забезпечують найкраще самопочуття та максимальну працездатність людини.

5.2. Аналіз шкідливих та небезпечних факторів

Згідно ГОСТ 12.0.002–80 ССБТ. «Термины и определения» умови праці визначаються сукупністю факторів виробничого середовища, які впливають на здоров'я і працездатність людини в процесі праці. Умовно ці чинники підрозділяються на небезпечні і шкідливі. Відповідно до ГОСТ 12.0.003 – 74 ССБТ «Опасные и вредные производственные факторы. Классификация». Вони діляться на чотири групи:

- фізичні;

- хімічні;
- біологічні;
- психофізіологічні.

Структурна схема класифікації небезпечних і шкідливих виробничих факторів (НШВФ) згідно ГОСТ 12.003 – 74 ССБТ наведена на рис. 5.2.



Рисунок 5.2 - Класифікація небезпечних і шкідливих виробничих факторів (НШВФ)

В основу цієї класифікації покладена природа їх дії на людину. Розглянемо фактори в тій послідовності, у якій вони наведені в нормативному документі.

До фізичних факторів відносяться: підвищена швидкість руху повітря; підвищена або понижена вологість; підвищений або понижений атмосферний тиск; недостатня освітленість; конструкції, що руйнуються; електричний струм; підвищений рівень статичної електрики; розміщення обладнання на висоті; підвищений рівень шуму і вібрації; гострі кромки обладнання тощо

(всього 14 факторів). Зрозуміло, що ці фактори проявляють фізичну дію на організм людини.

До хімічних факторів відносять: хімічні елементи, речовини та сполуки, що перебувають у різному агрегатному стані (твердому, газоподібному, рідкому); речовини, які різними шляхами проникають в організм людини, тобто через органи дихання, шлунково-кишковий тракт, шкірний покрив, слизові оболонки носа, рота і очей; речовини, які різко змінюють реактивність організму, тобто проявляють сенсibiliзуючу і алергічну дію на організм; речовини, які мають мутагенну дію або впливають на репродуктивну функцію людини.

Біологічні фактори розподіляються на патогенні (хвороботворні) мікроорганізми і макроорганізми.

Мікроорганізми проникають до організму людини у вигляді бактерій, вірусів, рикетсій (бактеріоподібні нерухомі мікроорганізми, які викликають гострі інфекційні захворювання), грибів і найпростіших.

Макроорганізми розподіляються на організми рослинного і тваринного походження. Вони можуть бути в харчовій сировині (зернові культури, бобові, соняшник тощо), в харчових виробництвах (хлібопекарському, цукровому, бродильному тощо) і є причиною захворювання працівників.

Психофізіологічні фактори розподіляються на фізичні і нервово-психічні перевантаження. До перших відносяться статичні, динамічні навантаження і гіподинамічні (обмежена рухова активність). Рівень статичного навантаження визначається величиною необхідного зусилля, часом його підтримки і положенням тіла працівника (позою) при виконанні роботи. Динамічні навантаження характерні при піднятті і переміщенні вантажів. Вони бувають легкі і важкі. Легкі відповідають підняттю вантажу масою до 5 кг з підлоги або переміщенню вантажу за зміну до 4 т, а важкі відповідно 40 кг і 6 т.

Нервово-психічні перевантаження розподіляються на розумове, перевантаження аналізаторів, монотонність праці і емоційні перевантаження. Розумове перевантаження є характерним для ІТ-спеціалістів при роботі

персональним комп'ютером. Та емоційне навантаження, яке також є характерним для спеціалістів при роботі з комп'ютером. Вони бувають мало напруженими, помірно напруженими, напруженими і дуже напруженими.

5.3. Розрахунок системи штучного освітлення приміщення відділу інформаційних технологій

Серед факторів зовнішнього середовища, що впливають на організм людини в процесі праці, світло займає одне з перших місць. Адже відомо, що майже 90% всієї інформації про довкілля людина одержує через органи зору. Під час здійснення будь-якої трудової діяльності втомлюваність очей, в основному, залежить від напруженості процесів, що супроводжують зорове сприйняття.

Світло впливає не лише на функцію органів зору, а й на діяльність організму в цілому. При поганому освітленні людина швидко втомлюється, працює менш продуктивно, зростає потенційна небезпека помилкових дій і нещасних випадків. Погане освітлення може призвести до професійних захворювань, наприклад, таких як робоча міопія (короткозорість), спазм акомодатії.

Для створення оптимальних умов зорової роботи слід враховувати не лише кількість та якість освітлення, а й кольорове оточення. Так, при світлому пофарбуванні інтер'єру завдяки збільшенню кількості відбитого світла рівень освітленості підвищується на 20-40% (при тій же потужності джерел світла), різкість тіней зменшується, покращується рівномірність освітлення.

При надмірній яскравості джерел світла та оточуючих предметів може відбутись засліплення працівника. Тому поверхні, що добре освітлюються і знаходяться в полі зору, краще фарбувати в кольори середньої світлості, коефіцієнт відбивання яких знаходиться в межах 0,3-0,6, і, бажано, щоб вони мали матову або напівматову поверхню.

Розрахунок системи штучного освітлення для відділу інформаційних технологій (ІТ) передбачає врахування ряду факторів, таких як площа

приміщення, тип робочих завдань, вимоги до рівня освітленості, види світлових джерел і їх характеристики.

Необхідно зробити розрахунок загального освітлення кабінету для ІТ спеціалістів. Довжина приміщення $A = 4$ м, ширина приміщення $B = 3$ м, висота $H = 3,5$ м. Висоти виробничої поверхні $h_p = 1$ м. Коефіцієнт відображення стелі $S_n = 70\%$, коефіцієнт відображення стін $S_c = 50\%$, робочої поверхні $S_p = 30\%$. Для освітлення приймаємо енергозберігаючі компактні люмінесцентні лампи, які не створюють пульсацій світлового потоку. Мінімальна освітленість світлодіодної лампи $E_{min} = 150$ лк. Напруга мережі 220В.

Для розрахунку загального освітлення кабінету для ІТ спеціалістів можна скористатися методом коефіцієнта використання світлового потоку. Розрахунок загального освітлення зазвичай включає визначення необхідної кількості світлового потоку для забезпечення заданої освітленості на робочій поверхні.

1. Розрахунок необхідного світлового потоку (Φ)

$$\Phi = \frac{E_{min} \cdot A \cdot B}{U \cdot K} \quad (5.1)$$

де E_{min} - мінімальна освітленість (150 лк);

A, B – ширина та довжина приміщення;

U - коефіцієнт використання світлового потоку;

K - коефіцієнт запасу (зазвичай від 1.5 до 2 для забезпечення достатнього рівня освітлення з урахуванням забруднення ламп та старіння).

2. Визначення коефіцієнта використання світлового потоку (U)

Коефіцієнт використання світлового потоку залежить від відношення висоти установки світильника до висоти приміщення, відображення стелі (S_n), стін (S_c) та робочої поверхні (S_p). Його можна визначити за допомогою спеціальних таблиць або калькуляторів.

3. Вибір світильників.

Після визначення необхідного світлового потоку вибираються світильники з відповідними характеристиками. Світловий потік одного світильника має бути вибраний так, щоб забезпечити рівномірне освітлення усієї площі приміщення.

Необхідний загальний світловий потік для забезпечення мінімальної освітленості 150 лк у кабінеті розміром 4 м на 3 м становить приблизно 1714 люмен. Нескладно підібрати єдиний світильник, що забезпечує потрібний світловий потік, але з урахуванням вимог до рівномірності освітлення краще використовувати конфігурацію з двох світильників меншої потужності.

5.4. Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів при роботі з персональним комп'ютером

5.4.1. Заходи щодо зменшення впливу

Україна має ряд нормативних актів, які стосуються організації робочого місця при роботі з персональним комп'ютером та вимагають дотримання ергономічних принципів для забезпечення здоров'я та безпеки працівників. Основними документами є Кодекс законів про працю України, а також Постанова Кабінету Міністрів України "Про затвердження Правил охорони праці при роботі з відображувальними екранами електронно-обчислювальних машин" від 18 січня 1995 року № 47.

- стаття 5 Кодексу законів про працю встановлює загальний принцип: "Кожен працівник має право на безпечні та нешкідливі умови праці".
- згідно зі статтею 212, роботодавець зобов'язаний забезпечити безпеку та нормальні умови праці, включаючи правильне організаційне та технічне обладнання робочого місця.
- постанова Кабінету Міністрів України № 47 визначає вимоги до організації робочого місця при використанні відображуваних екранів.
- зокрема, вони включають в себе вимоги до розташування моніторів, освітлення, розташування меблів та обладнання.

Згідно зі ст. 13 Закону України «Про охорону праці» роботодавець забезпечує організацію робіт з охорони праці на підприємстві. Для цього він:

- створює відповідні служби, у тому числі і службу з охорони праці;
- призначає посадових осіб і установлює їх обов'язки, права та відповідальність;
- організовує пропаганду безпечних методів праці та співробітництво з працівниками у галузі охорони праці;
- забезпечує впровадження прогресивних технологій, досягнень науки і техніки, позитивний досвід з охорони праці;
- забезпечує усунення причин, що приводять до нещасних випадків, професійних захворювань;
- забезпечує здійснення профілактичних заходів, визначених комісіями за підсумками розслідування нещасних випадків;
- організовує проведення аудиту охорони праці, атестації робочих місць на відповідність нормативно-правовим актам з охорони праці та вживає заходи по усуненню небезпечних і шкідливих виробничих факторів за їх підсумками;
- затверджує положення, інструкції, інші акти з охорони праці, що діють у межах підприємства;
- забезпечує належне утримання будівель і споруд, виробничого обладнання та устаткування, моніторинг їх технічного стану;
- забезпечує виконання необхідних профілактичних заходів відповідно до обставин, що змінюються.

Виробничі приміщення відповідно до вимог чинних нормативів мають бути забезпечені достатнім природним освітленням. Обов'язковим є також улаштування ефективної за екологічними і санітарно-гігієнічними показниками вентиляції. Висота виробничих приміщень повинна бути не менше 3,2 м, а об'єм і площа – 15 м³ та 4,5 м² відповідно на кожного працівника (для користувачів комп'ютерів згідно ДСанПіН 3.3.21007198

«Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» на одного працюючого повинно бути не менше: площі – 6 м² і об'єму – 20 м³).

Організація робочого місця згідно з ергономічними принципами повинна забезпечити зменшення впливу шкідливих та небезпечних факторів при роботі з персональним комп'ютером, таких як електромагнітне випромінювання, забезпечувати збереження здоров'я, підтримувати продуктивність праці, зберігати комфорт та безпеку співробітника. Важливо враховувати, що ергономіка — це індивідуальний аспект, і вимоги можуть різнитися в залежності від конкретних фізіологічних особливостей працівника.

Вимоги та рекомендації для організації ергономічного робочого місця для співробітників відділу інформаційних технологій наступні.

1) Забезпечити організацію робочого простору співробітника.

- а) робочий стіл та стілець повинні бути належної висоти, щоб уникнути напруження на ший та спині. Зазвичай, ширина столу повинна бути в межах 120-160 см. Це забезпечить достатньо простору для розташування комп'ютера, документів та інших робочих предметів. Оптимальна глибина столу зазвичай становить від 60 до 80 см. Це дозволяє забезпечити достатньо простору для робочої поверхні та зручно розташувати ноги. Висота столу повинна бути регульованою, або відповідати рівню ліктів працівника під час сидіння. Зазвичай це від 68 до 76 см. Важливо враховувати висоту робочого стільця, який також повинен бути регульованим;
- б) стілець повинен бути зручним, підтримувати природну кривизну хребта та мати можливість регулювання висоти;
- в) Ноги повинні стояти прямо на підлозі, а коліна повинні бути приблизно на одному рівні з тазом. Важливо, щоб під столом був достатній простір для ніг працівника. Глибина підстільної частини столу може бути від 45 до 60 см;

- г) Верхній край монітора повинен бути на рівні з очима або трошки нижче;
 - д) Монітор повинен бути розташований на відстані близько 20-30 дюймів (50-75 см) від очей.
 - е) Клавіатура розміщується на поверхні столу або висувній полиці на відстані 100-300 мм від краю, ближчого до користувача. Кут нахилу клавіатури має бути в межах 5-150. Поверхня клавіатури повинна бути матовою з коефіцієнтом відбиття 0,4. Клавiші клавіатури мають бути зручними в роботі і м'якими при натисканні (хід всіх клавiш має бути однаковим з мінімальним опором натискання 0,25 Н та максимальним – не більше 1,5 Н). Для уникнення травм запастися клавіатурою та мишею з ергономічним дизайном.
- 2) Освітлення і розташування робочих місць. Робоче місце користувача ПК необхідно розташовувати відносно світлових прорізів (вікон) так, щоб природне світло падало збоку, переважно зліва. При розміщенні робочих місць з ПК слід дотримуватися вимог, зазначених в НПАОП 0.00-1.28-10:
- а) Робочі місця розташовуються на відстані не менше 1 метра від стін зі світловими прорізами;
 - б) відстань між рядами робочих місць повинна бути не менше 1 метра;
 - в) відстань між бічними поверхнями відеотерміналів повинна бути не менш 1,2 метри;
 - г) відстань між тильною поверхнею одного відеотермінала й екраном іншого не повинна бути менш 2,5 метри;
- Вимоги двох останніх пунктів враховуються також при розміщенні робочих місць з ПК в суміжних приміщеннях з урахуванням конструктивних особливостей стін та перегородок.
- 3) Планування робочого часу і перерв. Державними санітарними правилами встановлюється такий режим праці й відпочинку при роботі з ПЕОМ при 8-годинній робочій зміні залежно від характеру праці:

- а) для розробників програм варто призначати регламентовані перерви для відпочинку тривалістю 15 хвилин щогодини;
- б) для операторів ПЕОМ варто призначати регламентовані перерви для відпочинку тривалістю 15 хвилин через кожні дві години;
- в) для операторів комп'ютерного набору варто призначати регламентовані перерви для відпочинку тривалістю 10 хвилин щогодини роботи з ПЕОМ.

У всіх випадках, коли виробничі обставини не дозволяють використати регламентовані перерви, тривалість безперервної роботи з ПЕОМ не повинна перевищувати чотирьох годин.

4) Підтримувати мікроклімат

Санітарні норми мікроклімату виробничих приміщень закріплені в ДСН 3.3.6.042-99 які встановлюють оптимальні і допустимі параметри мікроклімату залежно від загальних енерговитрат організму при виконанні робіт і періоду року.

За ступенем важкості роботи, що виконуються сидячи і не потребують фізичного напруження за своїм характером відносяться до «Легких робіт» категорії «Іа». Загальні енерговитрати організму дорівнюють 105–140 (90–120) Вт (ккал/год).

При санітарно-гігієнічному нормуванні умов виділяють два періоду року: теплий (середньодобова температура зовнішнього середовища вище +10С) і холодний (середньодобова температура зовнішнього середовища не перевищує 10С).

При виконанні робіт операторського типу, пов'язаних з нервово-емоційним напруженням в кабінетах, пультах і постах керування технологічними процесами, в залах обчислювальної техніки та інших приміщеннях повинні дотримуватися оптимальні умови мікроклімату для холодного періоду року: температура повітря 22-24 град. С, відносна вологість 60-40%, швидкість руху повітря не більш 0,1 м/сек. Умови

мікроклімату для теплого періоду року: температура повітря 23-25 град. С, відносна вологість 60-40%, швидкість руху повітря не більш 0,1 м/сек.

5.4.2. Техніка безпеки при роботі на комп'ютері

Основним нормативним документом прямої дії що регламентує улаштування електроустановок напругою до 1000В і вище, є «Правила улаштування електроустановок» (ПУЕ 2011). Виконання вимог ПУЕ є обов'язковим для всіх знов споруджуваних і реконструйованих електроустановок напругою до 500 кВ.

Вимоги до заземлення і захисних заходів електробезпеки систематизуються у главі 1.7 «Заземлення і захисні заходи електробезпеки», що затверджена Наказом Міністерства енергетики та вугільної промисловості України 31.03.2011 р. № 36, яка перероблена згідно з чинними в Україні вимогами стандартів, будівельних норм і правил, а також рекомендацій стандартів МЕК.

Технічна експлуатація електроустановок споживачів може здійснюватися за спеціальними правилами, установленими в галузі. Останні не повинні суперечити НПАОП 40.1-1.21-98 «Правила безпечної експлуатації електроустановок споживачів» та НПАОП 40.1-1.01-97 «Правила безпечної експлуатації електроустановок».

ПЕОМ і периферійні пристрої повинні підключатися до електромережі тільки за допомогою штепсельних з'єднань і розеток фабричного виконання, які повинні мати крім контактів фазного й нульового робочого провідників спеціальний контакт для підключення нульового захисного провідника.

Електромережі для живлення ПЕОМ варто виконувати за магістральною схемою по 3-6 штепсельні з'єднання або електророзеток в одному ланцюзі. Електророзетки на напругу 12 і 36 В по конструкції й колірному фарбуванню повинні відрізнятися від розеток на 127 або 220 В.

Електромережа розеток прокладається, як правило, у металевих трубах і гнучких металорукавах: уздовж стін приміщення з відводами відповідно до плану розміщення устаткування.

При розташуванні в приміщенні по його периметру до п'яти ПЕОМ допускається використовувати трьохжильний кабель в оболонці з негорючого або важкогорючого матеріалу без металевих труб або гнучких металевих рукавів.

Заземлені конструкції, що перебувають у приміщенні, (водопровідні труби, батареї опалення, тощо) повинні бути надійно захищені діелектричними щитками або сітками від випадкового дотику.

У приміщенні, де одночасно експлуатується або обслуговується більше п'яти ПЕОМ, на видному й доступному місці встановлюється аварійний резервний вимикач, яким можна повністю відключити електроживлення, за винятком освітлення.

Висновки до розділу 5

У п'ятому розділі, присвяченому охороні праці, розглядаються аспекти забезпечення безпечних і здорових умов праці, особливо важливі для запобігання травмам та захворюванням на робочому місці. Основна увага приділяється створенню умов, які сприяють продуктивній праці, збереженню здоров'я спеціалістів відділу інформаційних технологій. Підкреслюється, що забезпечення безпечних і комфортних умов, таких як освітлення, мікроклімат, ергономічне оформлене робочого місця праці і дотримання вимог безпеки позитивно впливає на загальний стан здоров'я працівників, а також підвищує їх продуктивність та ефективність.

Аналіз причин нещасних випадків на виробництві показує, що більшість з них в Україні пов'язані з організаційними недоліками, такими як слабе управління охороною праці, відсутність знань та навичок безпечної роботи, недостатній контроль за станом виробничого середовища, а також недоліки в правовому регулюванні трудових відносин.

П'ятий розділ також зосереджується на взаємозв'язку між безпекою працівника та його виробничим середовищем, яке включає об'єкти і фактори, породжені діяльністю людини, і які впливають на її здоров'я та самопочуття. Описано класифікацію небезпечних і шкідливих факторів, таких як фізичні, хімічні, біологічні та психофізіологічні, їх вплив на організм людини, а також зазначено необхідність зменшення цього впливу.

Розділ включає детальний опис розрахунку системи штучного освітлення приміщення для ІТ-спеціалістів, підкреслюючи значення якісного освітлення для здоров'я та продуктивності працівників. Описуються різні аспекти, які впливають на вибір освітлювальних приладів, включаючи розміри приміщення, тип робочих завдань та вимоги до освітленості.

Важливою частиною розділу є розробка заходів зі зменшення впливу шкідливих та небезпечних факторів при роботі з персональним комп'ютером. Обговорюються рекомендації для організації ергономічного робочого місця, включаючи розміщення обладнання, освітлення та планування робочого часу і перерв для зниження ризику професійних захворювань, таких як робоча міопія та напруження очей.

Завершується розділ детальним описом вимог до техніки безпеки при роботі на комп'ютері, зокрема вимог до заземлення, електробезпеки та захисних заходів, що необхідні для забезпечення безпеки працівників при роботі з електронним обладнанням.

6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1. Вступні положення

Охорона навколишнього середовища представляє собою комплекс заходів, спрямованих на ефективне використання та відновлення природних ресурсів, а також на захист унікальних природних комплексів і забезпечення екологічної безпеки. Вона включає різноманітні державні, адміністративні, правові, економічні та політичні ініціативи. Ці заходи орієнтовані на мінімізацію негативного впливу людської діяльності на природу.

Законодавча база охорони довкілля в Україні включає Закон України «Про охорону навколишнього природного середовища», а також низку інших законів, що регулюють використання землі, води, лісів, надр, атмосферного повітря, а також захист флори та фауни.

Головною метою охорони навколишнього середовища є запобігання та коригування шкідливих змін у довкіллі, які вже відбулися, відбуваються в даний час або можуть відбутися у майбутньому.

Природоохоронні заходи спрямовані на покращення екологічного стану середовища, включаючи підвищення екологічності виробництва, зниження споживання природних ресурсів, скорочення забруднення екосистем та зниження рівня шкідливих речовин у відходах.

Актуальність охорони довкілля на глобальному рівні посилюється через зростання антропогенного впливу, пов'язаного з демографічними змінами, урбанізацією, розвитком комунікацій, забрудненням довкілля та надмірним використанням природних ресурсів.

6.2. Забруднення навколишнього середовища

Згідно з даними американського Blacksmith Institute, Україна зіткнулася з однією з найбільш серйозних екологічних проблем серед світових держав. Згідно з офіційною статистикою, понад 70% території країни характеризуються умовами, небезпечними для життя людей, причому 1,7% від

усієї площі країни класифікуються як “території екологічного лиха”. Тільки 7% території України можна вважати відносно чистими, приблизно 8% земель вважаються умовно чистими, а 15% – малозабрудненими. Більшість цих менш забруднених територій знаходяться під охороною заповідників.

В індексі екологічних досягнень (Environmental Performance Index) за 2016 рік, Україна зайняла 44-е місце з 100 можливих балів, набравши 79,7 балів, що свідчить про вкрай негативну ситуацію в країні. Україна є лідером серед європейських держав за показниками обробки сільськогосподарських земель, використання водних ресурсів та вирубки лісів. Також країна стикається з серйозними проблемами повітряного забруднення в промислових регіонах, забруднення водойм і ґрунтів. Статус “зони екологічного лиха” позначається на інвестиційній привабливості України.

Незважаючи на те, що забруднення від промислових підприємств часто стає об'єктом громадської уваги через їх значний вплив на довкілля, важливо пам'ятати, що всі компанії, включно з ІТ-сектором, мають нести відповідальність за охорону навколишнього середовища. Навіть незначні, на перший погляд, джерела забруднення можуть мати кумулятивний ефект, особливо якщо мова йде про широкомасштабні галузі, такі як ІТ-індустрія.

Кожна компанія, незалежно від її розміру чи галузевої специфіки, має можливість впливати на довкілля, як позитивно, так і негативно. Відповідальне ставлення до екологічних питань передбачає не лише дотримання існуючих норм і законодавства, але й активне впровадження сталої практики у своїй діяльності. Це може включати використання енергоефективних технологій, мінімізацію відходів, використання відновлюваних джерел енергії, а також розвиток корпоративної екологічної культури.

Крім того, ІТ-компанії мають унікальну можливість використовувати свої технологічні ресурси для розв'язання екологічних проблем. Розробка програмного забезпечення для моніторингу стану довкілля, аналізу даних про

забруднення, розробка інноваційних рішень для зменшення впливу людської діяльності на природу - все це може мати значний позитивний вплив.

Таким чином, усвідомлення та сумлінне ставлення до екологічних питань повинні бути важливою частиною корпоративної стратегії кожної компанії. Це не лише сприяє захисту довкілля, але й покращує імідж компанії в очах споживачів, інвесторів та громадськості, створюючи позитивну репутацію як соціально відповідального бізнесу.

Розглянемо детальніше джерела забруднення навколишнього середовища компаніями ІТ - сектору.

1. Електронні відходи. ІТ-компанії часто мають справу з швидкими циклами оновлення обладнання. Відпрацьовані комп'ютери, монітори, принтери та інші електронні пристрої стають джерелом електронних відходів. Ці відходи містять різноманітні токсичні матеріали, такі як свинець, ртуть та кадмій, які можуть негативно впливати на навколишнє середовище, якщо їх не утилізувати належним чином. Важливо впровадити політику відповідального видалення та переробки цих відходів, щоб мінімізувати їх вплив на довкілля.

2. Енергоспоживання. ІТ-обладнання, особливо сервери, потребують значної кількості енергії для своєї роботи та охолодження. Хоча енергоспоживання невеликої ІТ-компанії може бути меншим, ніж у великих корпорацій, воно все одно має значний вплив. Використання енергозберігаючих технологій та перехід на відновлювані джерела енергії можуть допомогти знизити цей вплив.

3. Вплив на якість повітря всередині приміщення. Комп'ютерне обладнання може виділяти леткі органічні сполуки та інші забруднювачі, які погіршують якість повітря в офісі. Важливо забезпечити належну вентиляцію та чистоту в приміщеннях, де розміщене обладнання, для забезпечення здорового робочого середовища, як це було розглянуто в розділі 5.

4. Використання паперу та інших офісних матеріалів. Навіть у цифрову епоху ІТ-компанії часто використовують значні кількості паперу, картриджів для принтерів та інших офісних матеріалів. Зменшення використання паперу,

перехід на електронний документообіг та впровадження програм повторного використання та переробки можуть допомогти зменшити відходи.

5. Транспорт та викиди CO₂. Поїздки співробітників до офісу та відрядження сприяють викидам вуглекислого газу. Заохочення використання громадського транспорту, використання велосипедів або навіть впровадження політики віддаленої роботи можуть знизити цей вплив.

6.3. Розробка заходів щодо зменшення забруднення навколишнього середовища

Для мінімізації впливу на довкілля, ІТ-компанії можуть впровадити заходи екологічної відповідальності, такі як перехід на "зелені" технології, політику відповідального споживання та утилізації відходів, а також заохочення екологічно відповідальної поведінки серед співробітників.

Окрім того, доцільне напрацювання заходів та стратегій, що зменшують вплив кожного з чинників, що були розглянуті в попередньому розділі. Розглянемо деякі з таких заходів.

1. Поліпшення управління електронними відходами. ІТ-компанія може розвивати стратегії повторного використання та рециклінгу, співпрацювати з сертифікованими агентствами з утилізації для забезпечення правильного поводження з електронними відходами. Це може включати переробку компонентів для їх повторного використання в нових пристроях або безпечну утилізацію матеріалів, які не можна переробити.

2. Оптимізація споживання енергії. Цього можна досягати шляхом використання енергоефективного обладнання. Перехід на енергоефективні сервери, комп'ютери та інше офісне обладнання допоможе знизити загальне споживання електроенергії. Також доцільне впровадження систем управління енергоспоживанням. Автоматизоване управління енергоспоживанням у офісах та дата-центрах може значно знизити витрати енергії.

3. Зменшення використання паперу. Звичайно досягається шляхом переходу на безпаперові технології. Використання цифрових документів

замість паперових може значно знизити використання паперу. Оскільки неможливо повністю відмовитись від використання паперу, користь може також принести впровадження політики ефективного використання паперу, наприклад, двосторонній друк і використання переробленого паперу.

4. Екологічна свідомість і корпоративна культура. Цей пункт включає освітні програми для співробітників, що втілюється у підвищенні обізнаності серед співробітників про екологічні питання та впровадження практик, що сприяють сталому розвитку; та залученні співробітників до екологічних ініціатив: Наприклад, організація днів чистоти або участь у місцевих екологічних проектах.

5. Екологічно відповідальні бізнес-практики. Сюди відносяться: вибір постачальників і партнерів, які дотримуються екологічних стандартів, співпраця з компаніями, які також зобов'язані до принципів сталого розвитку, впровадження стандартів екологічної відповідальності в ланцюгах поставок.

Ці заходи не тільки сприяють зменшенню впливу на навколишнє середовище, але й можуть покращити репутацію компанії, залучити екологічно свідомих споживачів і знизити витрати завдяки енергозбереженню та ефективному використанню ресурсів.

Висновки до розділу 6

Шостий розділ, присвячений охороні навколишнього середовища, акцентує увагу на комплексі заходів, спрямованих на збереження природи та екологічну безпеку. Це включає різноманітні державні та приватні ініціативи, що охоплюють правові, економічні та політичні аспекти, метою яких є мінімізація негативного впливу людської діяльності на природу. Основна увага звернена на запобігання та коригування шкідливих змін у довкіллі, покращення екологічного стану та зниження рівня шкідливих речовин у відходах.

В розділі обговорюється проблематика забруднення навколишнього середовища в Україні, зокрема, високий рівень забруднення, що класифікує значну частину території країни як небезпечну для життя.

В контексті ІТ-компаній акцентується на важливості відповідального ставлення до охорони довкілля. Навіть невеликі компанії у сфері ІТ можуть сприяти забрудненню через електронні відходи, високе енергоспоживання, використання паперу та інших офісних матеріалів, а також через викиди CO₂, пов'язані з транспортом.

Розробка заходів щодо зменшення забруднення навколишнього середовища включає поліпшення управління електронними відходами, оптимізацію споживання енергії, зменшення використання паперу, підвищення екологічної свідомості та впровадження сталих бізнес-практик. Такі заходи не тільки сприяють охороні навколишнього середовища, але й покращують імідж компанії, створюючи позитивну репутацію як соціально відповідального бізнесу.

Важливо, щоб ІТ-компанії не лише дотримувалися існуючих норм та законодавства, але й активно впроваджували сталі практики у своїй діяльності, що включає використання енергоефективних технологій, мінімізацію відходів, рециклінг, використання відновлюваних джерел енергії та розвиток корпоративної екологічної культури. Це сприяє не лише зменшенню впливу на навколишнє середовище, але й є вигідним з точки зору економіки та репутації компанії.

ВИСНОВКИ

Роботу присвячено аналізу побудови систем підтримки прийняття рішень, можливим застосуванням таких систем в галузі управління нерухомістю та розробці програмного засобу, що втілює СППР на базі сучасних технологій, включаючи машинне навчання, штучні нейронні мережі та інноваційні технології програмування. Цілі дослідження включають підвищення ефективності традиційних СППР та їх адаптацію до сучасних викликів. Мета розробки програмного засобу - оптимізація процесів прийняття рішень в сфері нерухомості з акцентом на розширенні функціональності та продуктивності СППР, а також на інтеграції з сучасними технологіями для кращого аналізу даних та прогнозуванні.

Перший розділ фокусується на теоретичних аспектах функціонування СППР та значенні СППР у сфері нерухомості. Підкреслюється важливість СППР у ринковому аналізі, управлінні портфелем нерухомості, прогнозуванні та моделюванні, а також у візуалізації даних. Висвітлюється, як СППР може сприяти кращому розумінню ринкових трендів та цін, а також забезпечувати ефективне управління портфелем нерухомості та розробку стратегій інвестування. В подальшому розглядається концепція та класифікація СППР, з акцентом на процеси прийняття рішень. Розглядаються різні типи СППР, включаючи корпоративні та настільні системи, та їх особливості, такі як візуалізація даних та аналітична обробка, робиться висновок на важливості адаптації СППР до конкретних потреб користувачів. На базі розгляду концепції та архітектури СППР в подальших розділах проводиться конструювання та розробка СППР, що ґрунтується на алгоритмах машинного навчання та штучних нейронних мережах, використовує низку сучасних технологій для збільшення продуктивності розроблюваної СППР.

У другому розділі роботи розглядається моделювання та проектування системи підтримки прийняття рішень для управління нерухомістю, охоплюючи все від концепції до архітектури. Значна увага приділяється

створенню "проблемного процесору" та вводу-виводу через телеграм бот, включаючи вибір та обґрунтування регресійних та мовних моделей. Функціональні та нефункціональні вимоги визначаються через контекстну діаграму та сценарії використання, включаючи інтеграцію з чат-ботом, аналіз вартості нерухомості та її динаміки.

Архітектурно робота зосереджується на використанні мікросервісної архітектури, що забезпечує незалежність компонентів системи, таких як чат-бот та мікросервіси. Розробка включає планування управління даними, в тому числі розробку ER-діаграм для відображення структур зберігання інформації.

У третьому розділі роботи акцентується на важливості ретельного вибору технологій і інструментів для обробки та аналізу даних, а також на необхідності всебічного тестування для забезпечення якості програмного забезпечення. Значна увага приділяється вибору Python як основної мови програмування та FastAPI для створення мікросервісів, враховуючи їх продуктивність та спрощення розробки. Розглядаються також переваги використання PostgreSQL та SQLAlchemy для ефективної роботи з даними в Python, підкреслюючи їхню надійність та гнучкість. Важливість цих технологій у процесі розробки відображається через контрольні приклади та результати перевірки відповідності програмного засобу встановленим вимогам, що були сформульовані у попередньому розділі.

В одному з підрозділів розглянути приклади використання системи, що одночасно є контролем коректності реалізації та демонстрацією можливостей створеної системи. Також детально розглядається процес тестування розробленого програмного забезпечення, включаючи unit-тестування мікросервісів і використання моків для ізоляції тестів від зовнішніх залежностей. Оцінка покриття коду тестами висвітлює важливість якісного тестування для забезпечення надійності та стабільності розробленого ПЗ.

Ці аспекти разом відображають комплексний підхід до розробки та тестування програмного забезпечення, де використання сучасних

інструментів та методологій гарантує ефективність, гнучкість та високий рівень якості кінцевого продукту.

Четвертий розділ роботи зосереджений на оцінці економічної ефективності впровадження системи підтримки прийняття рішень у сфері нерухомості. Система, розроблена на базі мікросервісної архітектури з включенням машинного навчання, штучних нейронних мереж та мовних моделей, відрізняється унікальністю на ринку. Комерційний та технологічний аудит, проведений незалежними експертами, підтвердив високий комерційний потенціал системи завдяки інноваційності, конкурентним перевагам і оптимізації експлуатаційних витрат.

Економічна частина аналізу включає оцінку витрат на розробку та чистого прибутку від продукту, а також розрахунок періоду окупності. Виявлено, що загальні витрати на розробку становлять 865006 гривень, при цьому розрахунковий чистий прибуток та короткий період окупності близько 0,7 року свідчать про високу економічну ефективність та конкурентоспроможність розробки, що робить продукт привабливим для інвесторів та виробників.

П'ятий розділ розглядає аспекти охорони праці, акцентуючи на створенні безпечних умов праці для запобігання травм та захворювань на робочому місці. Особлива увага приділяється аналізу організаційних недоліків, що призводять до нещасних випадків, та необхідності зниження впливу шкідливих факторів, зокрема при роботі з персональним комп'ютером.

Розділ також зосереджується на важливості якісного освітлення та ергономічного організування робочих місць, що сприяє збереженню здоров'я та підвищенню продуктивності працівників. Деталізовано описано вимоги до техніки безпеки при роботі на комп'ютері, включаючи електробезпеку та захисні заходи.

У шостому розділі, який присвячений охороні навколишнього середовища, акцентується на комплексних заходах зі збереження природи та екологічної безпеки. Розглядається проблематика впливу ІТ-компаній на

навколишнє середовище, включаючи викиди CO₂, високе енергоспоживання та управління електронними відходами. Підкреслюється важливість впровадження сталих бізнес-практик у IT-секторі, включаючи оптимізацію споживання енергії, мінімізацію відходів, рециклінг та використання відновлюваних джерел енергії. Наголошується на необхідності розвитку корпоративної екологічної культури та активного впровадження енергоефективних технологій для зменшення негативного впливу на довкілля, що також сприяє підвищенню репутації та економічної ефективності компаній.

У даній роботі було проведено всебічний аналіз побудови та застосування систем підтримки прийняття рішень (СППР) у галузі управління нерухомістю, з акцентом на інтеграцію сучасних технологій, включаючи машинне навчання, штучні нейронні мережі, та інноваційні підходи до програмування. Робота виявила значний потенціал СППР у підвищенні ефективності управлінських процесів у сфері нерухомості, відкриваючи нові можливості для аналізу, прогнозування та стратегічного планування.

В цілому, робота демонструє комплексний підхід до розробки СППР, враховуючи сучасні технологічні тенденції та вимоги галузі управління нерухомістю.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Jones H., “An Essential Beginners Guide to Artificial Neural Networks and Their Role in Machine Learning and Artificial Intelligence”. CreateSpace Independent Publishing Platform, 2018, ISBN: 1647481023, стор. 76.
2. Teoh T., Rong Z., “Artificial Intelligence with Python”. Springer Nature, Singapore, 2022, ISBN: 9789811686153. 336 pages.
3. Tianqi C., Guestrin C., «XGBoost: A Scalable Tree Boosting System». The 22nd ACM SIGKDD International Conference, 2016. URL: <https://arxiv.org/abs/1603.02754>
4. Song T. та Hu T. «Research on XGboost academic forecasting and analysis modelling». The Second International Conference on Physics, Mathematics and Statistics, Hangzhou, China, 2019
5. Guolin K , Meng Q, Finley T., Wang T., Chen W., Ma W., Ye W., Liu T.-Y., «LightGBM: A Highly Efficient Gradient Boosting Decision Tree». Advances in Neural Information Processing System, № 30, 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf
6. Coelho L., Richert W., “Building Machine Learning Systems with Python”. Packt, 2018, ISBN: 1782161406. 290 pages.
7. Miles R., Hamilton K., “Learning UML 2.0: A Pragmatic Introduction to UML”, 1st edition, O’Reilly media, 2006, ISBN: 0596009828. 290 pages.
8. Michael Dawson, Programming with Python, Course Technology, 2020, ISBN: 9781435455009, 455 pages.
9. Zed Shaw, Learn Python 3 the Hard Way, Course Technology, 2017, ISBN: 9780134692883, стор. 320.
10. Python [Електронний ресурс]. Режим доступу: <https://docs.python.org/3/>- Назва з екрану.
11. FastAPI [Електронний ресурс]. Режим доступу: <https://fastapi.tiangolo.com/>- Назва з екрану.
12. Docker [Електронний ресурс]. Режим доступу: <https://www.docker.com/> - Назва з екрану.
13. PostgreSQL [Електронний ресурс]. Режим доступу: <https://www.postgresql.org/> - Назва з екрану.
14. Simon Riggs, Gianni Ciolli, PostgreSQL 14 Administration Cookbook, Packt Publishing, 2022, ISBN: 9781803248974, стор. 608.
15. Васильєв О., Програмування мовою Python, Богдан, 2020. ISBN: 9789661056113. Стор. 504.

16. Державні санітарні норми та правила «Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу». Наказ МОЗ України від 08.04.2014, № 248. Зареєстровано в Міністерстві юстиції України 6.05. 2014 р. за № 472/25249. <http://document.ua/>; <http://zakon.nau.ua/>.
17. ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень. [Електронний ресурс]. Режим доступу: <http://dnop.com.ua/>.
18. НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів. [Електронний ресурс]. Режим доступу: <http://dnop.com.ua/>.
19. Основи охорони праці: Підручник. 3-тє видання, доповнене та перероблене. / К. Н. Ткачук, В. В. Зацарний, Д. В. Зеркалов, О. І. Полукаров, В. С. Коз'яков, Л. О. Мітюк, Ю. О. Полукаров, Т. Є. Луц. За ред. К. Н. Ткачука - К.: Основа, 2014 - 456 с.
20. Основи законодавства України про охорону здоров'я: Закон України. – К., 1992. [Електронний ресурс]. Режим доступу: <http://zakon.rada.gov.ua/>.
21. Охорона праці в галузі інформаційних технологій: Навчальний посібник./ В.І. Голінько, М.Ю. Іконніков, Я.Я. Лебедєв – Дніпрпетровськ, НГУ, 2015 – 247 с.

ДОДАТОК А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра інформаційних управляючих систем та технологій

Спеціальність 124 "Системний аналіз"

Освітня програма "Системний аналіз"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

д.ф-м.н, проф. Ушкац М.В.

"_____" _____ 2023 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття ступеня вищої освіти "магістр"

Студенту _____ Золіну Олексію Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Системний аналіз методів та розробка СППР для управління та підбору об'єктів нерухомості

Керівник роботи:

Затверджені наказом ректора № 1098-уч. від "18" вересня 2023 року.

2. Термін подання роботи: 15 грудня 2023 р.

3. Вихідні дані до проекту (роботи): ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали наукового стажування

4. Перелік питань, що належать до розробки (найменування розділів):

1. Дослідження та аналіз методів підбору об'єктів нерухомості. Постановка задачі. 2. Розробка проекту СППР для управління та підбору об'єктів нерухомості. 3. Реалізація проекту СППР для управління та підбору об'єктів нерухомості. 4. Розрахунок економічної доцільності розробки СППР. 5. Охорона праці. 6. Охорона навколишнього середовища.

5. Перелік презентаційних матеріалів _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1			
2			
3			
4			
5			
6			

7. Дата видачі завдання 05 вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Вивчення і опис предметної області		
2	Розробка концепції системи		
3	Розділ з розробки проектних рішень по системі		
4	Реалізація проекту та розробка робочої документації		
5	Розрахунок економічної доцільності		
6	Вирішення питань охорони праці та навколишнього середовища		
7	Виконання графічних документів		
8	Оформлення пояснювальної записки		
9	Подання МКР на перевірку на плагіат		
10	Подання МКР рецензенту		
11	Подання МКР на державний захист		

Студент

(підпис)

Золін О.В.

(ПІБ)

Керівник роботи

(підпис)

Прізвище І.Пб.

(ПІБ)