

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА
НА ТЕМУ:
Система визначення оціночної вартості нерухомості

Галузь знань **12 «Інформаційні технології»**

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма **«Комп'ютерні науки»**

Освітній рівень: бакалавр

Виконала студентка 4 курсу, групи КН-41



Гринішак І. Д.

(прізвище та ініціали)

Керівник

Красовська Г. В.

(прізвище та ініціали)

кандидат технічних наук, доцент

(науковий ступінь, звання)

Випускна кваліфікаційна робота бакалавра допущена до захисту
рішенням кафедри *інтелектуальних технологій*

Протокол № _____ від _____ р.

зав. кафедри _____ доц. Іларіонов О. Є.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій
Кафедра інтелектуальних технологій
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ:

Зав. кафедри інтелектуальних технологій
доцент Іларіонов О. Є.
(звання, прізвище та ініціали)

(підпис)

«__» _____ 20__ р.

ЗАВДАННЯ

НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Гринішак Інні Дмитрівні

(прізвище, ім'я, по батькові)

1. Тема роботи

Система визначення оціночної вартості нерухомості

2. Термін здачі студентом закінченої роботи

28 травня 2023 року

3. Вихідні дані до роботи: нормативно-правова база оцінки нерухомості США; статті про нерухомість та процес її оцінки; наукові джерела про багатокритеріальну оптимізацію.

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити): вступ; розділ №1: актуальність задачі оцінки нерухомості; особливості процесу оцінки нерухомості в США (нормативно-правова база, оціночні критерії, підходи до визначення оцінки); порівняльний аналіз систем-аналогів; розділ №2: автоматизовані методи розрахунку оцінки нерухомості; функціональний аналіз системи; проектування архітектури системи та інформаційного забезпечення системи; розділ №3: реалізація програмного забезпечення; керівництво користувачів системи; процес тестування; висновки.

5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій):

Вступ – 2 та 3 слайд; перший розділ – 4...7 слайди; другий розділ – 8 ... 12 слайди; третій розділ – 12 ... 21 слайди; висновки – 22 слайд.

6. Консультація з випускної кваліфікаційної роботи із зазначенням її розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання

13 лютого 2023 року

Керівник _____ / Красовська Г. В. /
(підпис) (ініціали та прізвище)

Завдання прийняв до виконання _____ / Гринішак І. Д. /
(підпис) (ініціали та прізвище)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
1	Аналіз особливостей задачі оцінки майна	13.02.2023 – 26.02.2023	Виконано
2	Дослідження методів оцінки нерухомості	26.02.2023 – 03.03.2023	Виконано
3	Аналіз систем-аналогів, визначення вимог	03.03.2023 – 05.03.2023	Виконано
4	Проведення функціонального аналізу	05.03.2023 – 10.03.2023	Виконано
5	Побудова архітектури системи	10.03.2023 – 15.03.2023	Виконано
6	Проектування інформаційного забезпечення	15.03.2023 – 30.03.2023	Виконано
7	Розробка програмного забезпечення	30.03.2023 – 30.04.2023	Виконано
8	Формування керівництва користувачів	30.04.2023 – 05.05.2023	Виконано
9	Тестування розробленої системи	05.05.2023 – 10.05.2023	Виконано
10	Побудова пояснювальної записки та висновків	10.05.2023 – 20.05.2023	Виконано
11	Підготовка доповіді та презентації	20.05.2023 – 28.05.2023	Виконано

Студент _____ / Гринішак І. Д. /
(підпис) (ініціали та прізвище)

Керівник випускної кваліфікаційної роботи _____ / Красовська Г. В. /
(підпис) (ініціали та прізвище)

Анотація

Гринішак Інна Дмитрівна виконала випускню кваліфікаційну роботу на тему «Система визначення оціночної вартості нерухомості» за спеціальністю 122 — «Комп'ютерні науки».

У випускній кваліфікаційній роботі проведено аналіз сучасних систем визначення оціночної вартості нерухомості, запропоновано новий алгоритм вирішення задачі оцінки нерухомості за допомогою багатокритеріальної оптимізації, а саме методу аналізу ієрархій, розроблено інформаційне і програмне забезпечення, що виконує процедуру визначення оціночної вартості нерухомості. У цьому дослідженні також були визначені цікаві задачі, пов'язані з оцінкою нерухомості, які не були вирішені існуючими системами на момент проведення дослідження. Для цих задач розроблено необхідні підсистеми.

Ключові слова: оціночна вартість, нерухомість, багатокритеріальна оптимізація, метод аналізу ієрархій, система.

Summary

The degree project «System for determining the estimated value of real estate» has been completed by **Hrynishak Inna** specialty 122 — «Computer Science».

In the graduation thesis, an analysis of modern systems for determining the appraised value of real estate was carried out, a new algorithm for solving the problem of real estate appraisal using multi-criteria optimization, namely the method of analyzing hierarchies, was proposed, and information and software was developed that performs the procedure for determining the appraised value of real estate. This study also identified interesting problems related to real estate valuation that were not solved by the existing systems at the time of the study. Necessary subsystems have been developed for these tasks.

Keywords: appraised value, real estate, multi-criteria optimization, method of analysis of hierarchies, system.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	5
ВСТУП.....	6
РОЗДІЛ 1. АНАЛІЗ ПРОЦЕСУ ОЦІНКИ НЕРУХОМОСТІ.....	9
1.1 Актуальність задачі оцінки нерухомості	9
1.2 Нормативно-правова база оцінки нерухомості	11
1.3 Фактори, що впливають на оцінку нерухомості	13
1.4 Підходи до визначення оціночної вартості нерухомості	16
1.5 Узагальнена технологія оцінки об'єкту нерухомості	19
1.6 Аналіз систем-аналогів, що розраховують оцінку нерухомості	20
1.7 Формулювання вимог до системи для оцінки нерухомості.....	22
1.8 Висновок.....	25
РОЗДІЛ 2. ПРОЕКТНІ РІШЕННЯ ДЛЯ СИСТЕМИ ВИЗНАЧЕННЯ	
ОЦІНОЧНОЇ ВАРТОСТІ НЕРУХОМОСТІ.....	27
2.1 Проектні вирішення задачі оцінки нерухомості	27
2.2 Функціональний аналіз системи оцінки нерухомості	38
2.3 Архітектура системи для визначення оцінки нерухомості.....	40
2.4 Проектування бази даних для розроблюваної інформаційної системи.	43
2.5 Висновок.....	46
РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ВИЗНАЧЕННЯ	
ОЦІНОЧНОЇ ВАРТОСТІ НЕРУХОМОСТІ.....	47
3.1 Обґрунтування вибору програмних засобів	47
3.2 Структура програмного забезпечення.....	50
3.3 Керівництво користувачів системи визначення оцінки нерухомості....	58
3.4 Тестування системи визначення оціночної вартості нерухомості	68
3.5 Висновок.....	82
ВИСНОВОК	83
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	84
Додаток А	87
Додаток Б	90

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ОН — об'єкт нерухомості

МАІ — метод аналізу ієрархій

БД — база даних

ПЗ — програмне забезпечення

ВСТУП

Актуальність дослідження. Кожна сучасна людина замислюється про спосіб вигідних інвестицій власних коштів, щоби зберегти їх від згубного впливу інфляції та мати можливість отримати додатковий прибуток. Досягти зазначених вище цілей допоможуть зважені інвестиції в нерухомість. Однак, при виборі інвестиційного об'єкту нерухомості (ОН) потрібно ретельно вивчити його характеристики, ризики та перспективи отримання прибутку. Отже, потенційному інвестору для успішного інвестування необхідно самому мати достатній рівень знань та досвіду або звертатися до фахівців в області оцінювання нерухомого майна для проведення всебічного аналізу варіантів інвестування і вивчення різних факторів і обставин, які можуть вплинути на вартість майна. Таким чином, створення системи визначення оцінки ОН є дуже актуальним та затребуваним питанням.

Основною метою є проектування та реалізація автоматизованої системи визначення оціночної вартості нерухомості, що забезпечить потенційних інвесторів в нерухомість зручним та ефективним інструментарієм, який допоможе їм заощадити час та гроші при виборі вдалих інвестиційних ОН.

Об'єкт дослідження: процес оцінки об'єкта нерухомості.

Предмет дослідження: метод аналізу ієрархій (MAI) для оцінювання об'єктів.

Завдання дослідження:

- Дослідити особливості задачі оцінки нерухомості:
 - Проаналізувати нормативно-правову базу США щодо оцінки нерухомості;
 - Визначити фактори, що впливають на оцінку;
 - Описати підходи до визначення оціночної вартості ОН;
 - Узагальнити технологію оцінки нерухомості;
- Провести детальний аналіз систем-аналогів, що розраховують оцінку ОН, визначити їх плюси та мінуси;

- Сформулювати вимоги до розроблюваної системи для оцінки ОН;
- Дослідити методи вирішення задачі оцінки нерухомості такі як регресійний аналіз, алгоритми машинного навчання та багатокритеріальну оптимізацію. Для останнього, описати основні методи та описати їх алгоритми. В результаті дослідження обрати спосіб вирішення задачі;
- Провести функціональний аналіз системи оцінки нерухомості;
- Спроекувати та реалізувати підсистеми, що забезпечують інформаційну базу системи для проведення оцінки об'єкта нерухомості:
 - Спроекувати та реалізувати базу даних (БД) інформаційної системи;
 - Спроекувати та реалізувати архітектуру — серверну та клієнтську сторону системи визначення оцінки ОН;
- Навести керівництво користувачів розроблюваної системи;
- Провести тестування розробленого застосунку.

В *першому розділі роботи* досліджено особливості процесу оцінки нерухомості, що включає: аналіз нормативно-правової бази США, аналіз факторів, що впливають на оцінку та виокремлення найважливіших з них. Даний розділ включає в себе опис методик визначення оціночної вартості об'єкта нерухомості та узагальнену технологію оцінки об'єкта. Окрім згаданого, в розділі наведено порівняльний аналіз існуючих систем-аналогів, що дозволило сформулювати задачу роботи та вимоги до неї.

В *другому розділі* описано методи, які можуть вирішити задачу оцінки нерухомості. Більш детально розглянуто МАІ та запропоновано алгоритм вирішення задачі роботи з точки зору цього методу. Окрім згаданого, в даному розділі проведено функціональний аналіз розроблюваної системи визначення оціночної вартості нерухомості, та спроекувано підсистеми, що забезпечують інформаційну базу системи (база даних, серверна та клієнтська сторона системи).

Третій розділ роботи включає опис та обґрунтування програмних засобів, що були використані при реалізації розробленої системи. В цьому розділі надано докладний опис структури програмного забезпечення системи, включаючи структурні схеми та специфікації його програмних модулів. В кінці цього розділу наведено керівництво для користувачів системи, а також описано процес тестування системи з метою впевненості в правильній роботі системи і відповідності всім вимогам, які були поставлені на початку роботи.

В результаті виконання роботи було спроектовано та розроблено систему визначення оціночної вартості нерухомості за допомогою методу аналізу ієрархій. Було доведено її актуальність, описано процес проектування та процес реалізації, а також перевірено коректність її роботи. Щодо *практичного значення одержаних результатів*, то дану систему можна використати для прогнозування ціни об'єкта нерухомості, з метою забезпечення розумної інвестиції заощаджень та попереднього аналізу всіх ризиків. Даний продукт буде корисним та необхідним в першу чергу людині, яка хоче зробити будь-які маніпуляції з об'єктами нерухомості, або ж для компаній, які займаються продажом і викупом нерухомості.

РОЗДІЛ 1. АНАЛІЗ ПРОЦЕСУ ОЦІНКИ НЕРУХОМОСТІ

1.1 Актуальність задачі оцінки нерухомості

У сучасному світі люди все більше задумуються про вигідні способи інвестування своїх коштів. Головна мета полягає в збереженні вкладень від негативного впливу інфляції. Крім того, важливо, щоб інвестовані кошти працювали та приносили додатковий прибуток. Досягти зазначених вище цілей допоможуть інвестиційні інструменти, використовувані на тривалому часовому проміжку. При цьому важливо, щоб вони мали мінімальний ризик і були вельми перспективними. Саме такими і є інвестиції в нерухомість [1].

Необхідність у житлі у людини виникла багато років тому і зберіглася донині. Нікуди не дінеться вона і в майбутньому. Тому нерухомість завжди буде користуватися попитом, а значить, вона є відмінним інструментом для інвестування коштів. Більше того, такі вкладення являють собою цілком прийнятний варіант для ведення бізнесу. Для цього зовсім не обов'язково мати величезні грошові суми. Вкладати гроші в нерухомість можна ще на початковому етапі будівництва. Крім того, є можливість стати учасником житлового кооперативу, купивши в ньому частку.

В даній роботі буде проаналізовано ринок США, через наступні причини[2]:

1. Американський закон не обмежує покупку іноземцями житлових і комерційних об'єктів.
2. В США, не дивлячись на світову кризу, економіка залишається на стабільному рівні. Звідси випливає, що якщо в США купити нерухомість, то в майбутньому можна отримати вигоду з її продажі.
3. Сполучені Штати Америки багаті на різноманітність клімату і природи. В США можна вибрати найрізноманітніше житло, починаючи від вілли на березі моря і закінчуючи квартирою де-небудь в південному жаркому штаті. Любителі гір можуть придбати житло біля підніжжя найкрасивіших вершин. До того ж гарантією високого рівня екології та

безпеки в цій країні, виступає закон США про захист навколишнього середовища.

4. Якість житла. Всі будівництва в штатах контролюються на всіх етапах будівництва муніципальними органами, і виробляються тільки тими організаціями, які мають потрібну ліцензію.

Проте як і у будь-якого іншого фінансового інструменту у інвестицій в нерухомість є як плюси, так і мінуси.

Серед переваг такого виду вкладень можна виділити наступні:

- нерухомість має високу ліквідність;
- на тривалому періоді незмінна прибутковість, наприклад, здаючи нерухомість в оренду, можна отримувати прибуток довгі роки;
- відносна доступність вкладень;
- широкий вибір варіантів для інвестування.

Незважаючи на суттєві переваги вкладень у нерухомість, всі існуючі варіанти інвестування схильні до ризику.

Основними недоліками таких інвестицій є:

- попит на нерухомість знаходиться в досить відчутній залежності від економічної обстановки в країні в цілому і окремому регіоні зокрема;
- ціни на об'єкти нерухомості (ОН) досить високі;
- у маленьких містах попит на нерухомість знаходиться на досить низькому рівні;
- високі додаткові витрати — комунальні, ремонт, податки.

Більш того, існує можливість форс-мажорів. Трапляється, що ціна на об'єкт нерухомості різко падає в силу непереборних обставин. Наприклад, квартири в екологічно чистому районі подешевшають, якщо поблизу побудують завод або автодорогу. У результаті інвестор не тільки нічого не заробить, але і може втратити частину вкладених грошей.

Щоб уникнути більшості проблем, до моменту інвестування важливо провести попередній аналіз. В ході нього порівнюються можливі варіанти для

вкладення коштів і вивчаються різні фактори і обставини, які можуть вплинути на вартість нерухомості.

Саме проведення попереднього аналізу об'єкта нерухомості і є цікавою задачею для реалізації. Зробити продукт, який на основі якісних та кількісних факторів зможе спрогнозувати оціночну вартість об'єкта нерухомості, саме за допомогою такого ресурсу користувач зможе зробити висновок, чи досліджуваний об'єкт нерухомості хороший варіант для інвестицій чи ні. Даний продукт буде корисним та необхідним в першу чергу людині, яка хоче зробити будь-які маніпуляції з об'єктами нерухомості, або ж для компаній, які займаються продажем і викупом нерухомості, адже система дасть можливість збільшити кількість можливих угод завдяки врахування факторів ризику та часу, а також надання користувачеві зручного інструментарій з можливістю експериментувати з вхідними даними, що гарантує знаходження вдалого об'єкту для продажу чи покупки.

1.2 Нормативно-правова база оцінки нерухомості

Для створення системи, яка може оцінити нерухомість, необхідно врахувати нормативно-правову базу США. Ці правові норми впливають на визначення оціночних критеріїв, методів оцінки і т.д. Дотримання вказаних норм і вимог є важливим елементом забезпечення якості і точності оцінки нерухомості в США.

Нормативно-правова база оцінки нерухомості у США складається зі значної кількості законів, правил, нормативних документів та стандартів, які визначають вимоги до процесу оцінки нерухомості.

Основні законодавчі документи, які стосуються оцінки нерухомості в США, такі:

- Федеральний закон про оцінку нерухомості (FIRREA [3]), що був прийнятий у 1989 році. Цей закон створив Федеральну комісію з оцінки (Appraisal Subcommittee) та встановив національні стандарти

оцінки нерухомості, які визначають вимоги до професійної компетентності оцінювачів, методології оцінки та звітності.

- Закон про збереження фінансової стійкості (Dodd-Frank Act [4]), прийнятий в 2010 році, встановлює нові вимоги до оцінки нерухомості, зокрема, щодо вимог до повноважень оцінювачів та забезпечення незалежності від сторін угоди.
- Додаток до федерального регулюючого документа (Federal Register) з встановлення стандартів оцінки нерухомості (USPAP [5]), що розробляється і підтримується Федеральною комісією з оцінки. USPAP визначає мінімальні вимоги до методології, звітності та етики при оцінці нерухомості, та є обов'язковими до застосування професійними оцінювачами на всіх рівнях діяльності: федеральному, у штатах та на муніципальному рівні.

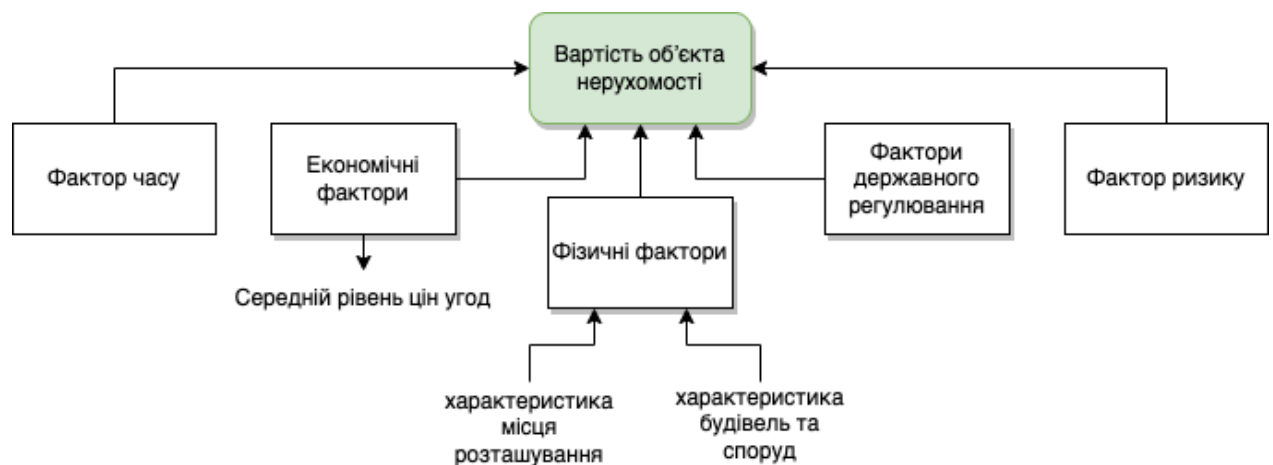
Разом з єдиними стандартами, поточне регулювання оцінки нерухомого майна ґрунтується на законодавчих актах відповідного штату, тому вибір штату є дуже важливим критерієм при виборі об'єкта нерухомості, бо відрізняються вони не тільки кліматом, але й соціально економічними умовами, що істотно впливає на вартість нерухомого об'єкта. На законодавчому рівні для штату зазначено уточнення, особливості для проведення оцінки в ньому [6].

Також на рівні штату закріплено список допустимих ліцензованих та зареєстрованих експертів(також їх називають ще агентами нерухомості або брокерами). У Сполучених Штатах Америки лістинг нерухомості, тобто договір з інформацією про нерухомість (характеристика, ціна та інші деталі), передається ліцензованим брокерам, які використовують єдину систему MLS та кілька сервісів для перевірки об'єктів. У американських експертів нерухомості обов'язково має бути ліцензія для роботи. Законодавство про підтвердження кваліфікації таких фахівців у кожному штаті відрізняється, однак скрізь має значення рівень освіти, тип та глибина атестаційних іспитів брокера нерухомості. Вони виступають як оцінювачі, що на основі поданих їм даних та

чинного законодавства вираховують коефіцієнт, який використовується для отримання оцінки розглянутої нерухомості.

1.3 Фактори, що впливають на оцінку нерухомості

Згідно загальних стандартів зміна вартості об'єкта нерухомості залежить від низки факторів, які наведено на рисунку 1.1.



Рисунк 1.1 — Групи факторів, що впливають на оцінку нерухомості

Під впливом економічних факторів формується середній рівень цін угод. До економічних факторів належать: економічний рівень розвитку штату, грошово-кредитна політика та інші. Економічними факторами, що визначають попит, є: зайнятість населення, рівень середньої заробітної плати та доходів населення, рівень цін, наявність джерел фінансування, доступність та умови кредиту на купівлю житла. На пропозицію впливають: наявність вільних земельних ділянок та об'єктів, що простоюють; орендні ставки та ціни на готові об'єкти; собівартість будівництва. Вся ця інформація є у доступі кваліфікованого брокера, який приймає роль при розрахунку оцінки.

Фізичними факторами є: характеристика місця розташування, враховується рівень розвитку конкретного регіону, міста, характер сусіднього оточення або престижність району, відстань від центру, ступінь розвитку

інфраструктури; характеристики будівель та споруд: тип будівлі, рік побудови, стиль, стан будівлі (необхідність капітального ремонту) тощо.

Щодо фізичного фактору, який визначає тип будівлі, то у США є 5 основних типів нерухомості [7]:

- Окремий будинок (Single-family home): житлові будинки з однією квартирою, призначені для проживання однієї сім'ї.
- Багатоквартирний будинок (Multifamily home): житлові будинки з кількома окремими квартирами, призначені для проживання кількох сімей. Це можуть бути апартamenti, кондомініуми, таунхауси та інші типи житла.
- Квартира в багатоквартирному будинку (Apartment): окрема квартира в багатоквартирному будинку, яка зазвичай здається в оренду.
- Кондомініум (Condominium): окрема квартира або будинок, який знаходиться у власності особи, але має спільну власність на спільні приміщення та ділянки з іншими власниками квартир чи будинків.
- Таунхаус (Townhouse): житловий будинок з кількома поверхами, який має спільний стіну з сусідніми будинками, але має власний вхід та приватний двір.

З усіх груп факторів, під впливом яких формується вартість об'єктів, особливо слід виділити фактор часу та ризики. Фактор часу, мабуть, найважливіший фактор, що впливає на всі ринкові процеси, на ціну, на вартість, на прийняття рішень. Ринкова вартість об'єкта змінюється в часі, під впливом багатьох факторів, тому вона може бути розрахована на даний момент часу. Завтра ринкова вартість може бути вже іншою. Отже, постійна оцінка і переоцінка об'єктів власності в цілому є необхідною в умовах ринкової економіки. При оцінці важливо пам'ятати, що існують різні види ризику, і жодне з вкладень в умовах ринкової економіки не є абсолютно безризиковим. Під ризиком розуміють непостійність і невизначеність, пов'язані з кон'юктурою ринку, макроекономічними процесами тощо. Іншими словами, ризик — це ймовірність того, що доходи, які будуть отримані від інвестицій, виявляться

більші або менші від прогнозованих. Усі види ризиків так чи інакше мають фінансовий вимір, збільшуючи витрати за проектом або угодою та знижуючи дохід, або призводять до перевищення запланованих строків інвестування, або у крайньому випадку — до втрати не тільки доходу, але й капіталу. Тому необхідно шукати всі можливі шляхи для зменшення ступеню ризиків.

Серед великої кількості факторів, необхідно виділити основні, що найбільше впливають на вартість нерухомості. Після аналізу статті "What Factors Determine Real Estate Prices?" [8], виділимо наступні показники (рисунок 1.2), які необхідні для розрахунку оцінки нерухомості, а саме квартири:



Рисунок 1.2 — Необхідні фактори-показники для розрахунку оцінки

Більшість показників може бути надано людиною, яка хоче продати / купити нерухомість. Однак, для отримання оцінки якісних показників, таких як рейтинг штату, рейтинг міста тощо, необхідно звернутися до кваліфікованого експерта з нерухомості або використовувати перевірені зовнішні ресурси. На основі цієї інформації можна проаналізувати фактор ризику, який враховуватиметься під час кінцевого оцінювання нерухомості.

1.4 Підходи до визначення оціночної вартості нерухомості

Визначивши критерії, за якими буде здійснюватися оцінка, необхідно вибрати яким способом з них буде отримана оцінка. Існують три традиційні групи методик визначення вартості об'єкта нерухомості [9]:

- Підхід порівняння продажів (порівняння характеристик нерухомості, яка оцінюється, з схожим нещодавно проданим об'єктом нерухомості).
- Витратний підхід (покупець не заплатить за нерухомість більше, ніж коштувало б будівництво еквівалента).
- Дохідний підхід (подібний до методів, що використовуються для фінансової оцінки, аналізу цінних паперів або ціноутворення облігацій).

При обчисленні оцінки потрібно аналізувати, хто покупець і який тип нерухомості ним розглядається, а також яка інформація є доступною для роботи. Це, як правило, спрямовує на вибір найкращого методу оцінки для конкретної ситуації. Наприклад, оцінка нерухомості, яку зазвичай купують бізнесмени (хмарочоси, офісні будівлі і т.д.), може розраховуватися за допомогою дохідного підходу. Покупці, зацікавлені в придбанні житлової нерухомості для однієї сім'ї, оберуть підхід аналізу ринку. Третій і останній підхід — це витратний підхід для обрахунку вартості є найбільш корисним для визначення страхової вартості та вартості будівництва нової споруди чи будівлі.

1.4.1 Підхід порівняння продажів

Даний підхід припускає, що розсудлива (або раціональна) особа буде порівнювати ціни на ринку та буде прагнути придбати нерухомість, яка відповідає її бажанням і потребам, за найнижчою ціною.

Щодо процесу оцінки, то спочатку збираються дані про нещодавні продані об'єкти нерухомості, які подібні до об'єкта, для якого обчислюється оцінка. Лише продане майно може використовуватися для визначення вартості майна,

оскільки воно представляє суми, фактично сплачені або узгоджені за нерухомість. Ці дані можна знайти в публікаціях про нерухомість, публічних записах покупців, продавців, брокерів та/або агентів з нерухомості, тощо. Важливі деталі кожного об'єкту нерухомості описані у звіті про оцінку. Оскільки продані об'єкти, які аналізується, не є ідентичними об'єкту нерухомості над яким ведеться оцінка, потрібно внести коригування щодо дати продажу/покупки, місця розташування, стилю, зручностей, квадратних метрів, розміру ділянки тощо. Основна ідея полягає в тому, щоб змодельовати ціну, яку було б сплачено, на основі фактичної вартості за уже проданий схожий об'єкт, але з урахуванням експертного коригування.

Етапи підходу порівняння продажів:

1. Дослідити ринок, щоб отримати інформацію про продажі подібні до об'єкта нерухомості, який оцінюється.
2. Визначити відповідні одиниці порівняння і розробити порівняльний аналіз для кожної з них.
3. Виконати коригування факторів.
4. Узгодити численні показники вартості, які є результатом коригування, у єдиний показник вартості.

1.4.2 Витратний підхід

Витратний підхід, або підхід підсумовування, полягає в тому, що вартість нерухомості можна оцінити шляхом підсумовування вартості землі та амортизованої вартості будь-яких покращень [10]. Вартість удосконалень часто позначається аббревіатурою RCNLD (що означає «вартість відтворення/заміни новим без амортизації»). Відтворення означає відтворення точної копії; вартість заміщення відноситься до вартості будівництва будинку або іншого поліпшення, яке має таке саме значення, але з використанням сучасного дизайну, якості виготовлення та матеріалів. На практиці, оцінювачі майже завжди використовують вартість відновлення, а потім віднімають коефіцієнт для будь-

якої функціональної некорисності, пов'язаної з віком об'єкта власності. Винятком із загального правила використання вартості відновлення є деякі оцінки страхової вартості. У таких випадках метою є відтворення точного активу після руйнівної події, наприклад пожежі.

У більшості випадків, коли використовується витратний підхід, загальна методологія є гібридом підходів до порівняння витрат і продажів (відображаючи як витрати постачальників, так і ціни, які шукають клієнти). Наприклад, відновну вартість будівництва будівлі можна визначити шляхом додавання витрат на оплату праці, матеріалів та інших витрат. З іншого боку, вартість землі та амортизація повинні бути отримані з аналізу порівнянних даних про продажі. Витратний підхід вважається найбільш надійним при використанні на нових структурах, але метод стає менш надійним для старих об'єктів.

1.4.3 Дохідний підхід

Підхід капіталізації доходу (його часто називають просто «підходом доходу») використовується для оцінки комерційної та інвестиційної нерухомості. Оскільки він призначений для прямого відображення або моделювання очікувань і поведінки типових учасників ринку, цей підхід зазвичай вважається найбільш застосовним методом оцінки нерухомості, яка приносить дохід, якщо існує достатня кількість ринкових даних.

У комерційній нерухомості, яка приносить дохід, цей підхід капіталізує потік доходу в показник вартості. Це можна зробити за допомогою мультиплікаторів доходу або ставок капіталізації, застосованих до чистого операційного доходу (NOI). Чистий операційний прибуток (NOI) — це валовий потенційний дохід (GPI) за вирахуванням вакансій і втрат від збору (= ефективний валовий дохід) за вирахуванням операційних витрат (за винятком обслуговування боргу, податків на прибуток та/або амортизаційних відрахувань, що застосовуються бухгалтерами).

1.5 Узагальнена технологія оцінки об'єкту нерухомості

Описані вище методики для оцінки нерухомості є досить точними, але вони мають багато серйозних мінусів: по-перше, дані методики є затратними та дорогими; по-друге, вони не дають відповідь одразу, на виконання всіх кроків того чи іншого методу потрібно багато часу.

Для вирішення вказаних проблем, пропонується використання наукової методології оцінки, іншими словами автоматизованих моделей оцінки (AVMs) [11].

Зазвичай, виділяють наступні етапи даного процесу:

1. Збір даних: збір даних про ОН, що оцінюється, включає в себе збір факторів, що описані в 1.3. Дані надаються користувачем, а також збираються з зовнішніх ресурсів (або надаються експертом нерухомості).
2. Підготовка даних: підготовка даних включає в себе проведення аналізу даних, щоб визначити, які фактори мають найбільший вплив на вартість нерухомості.
3. Вибір моделі: вибір моделі залежить від того, яка інформація доступна та який тип житлової нерухомості оцінюється.
4. Розробка моделі: розробка моделі включає в себе визначення статистичних методів та алгоритмів, що будуть використовуватись для оцінки житлової нерухомості. Це може включати в себе використання регресійних моделей, методів багатокритеріальної оптимізації, нейронних мереж або інших методів машинного навчання.
5. Калібрування моделі: калібрування моделі включає в себе перевірку точності та надійності моделі шляхом порівняння прогнозування ринкової вартості з реальними продажами нерухомості.
6. Валідація моделі: після калібрування моделі проводиться валідація, щоб переконатися в її точності та надійності.

1.6 Аналіз систем-аналогів, що розраховують оцінку нерухомості

Перед формулюванням вимог до системи, варто проаналізувати системи-аналогі, які надають засоби для розрахунку оцінки, визначити їх плюси та мінуси.

В США є декілька систем, які можуть запропонувати автоматизований розрахунок оцінки нерухомості, розглянемо найвідоміші з них (рисунок 1.3).

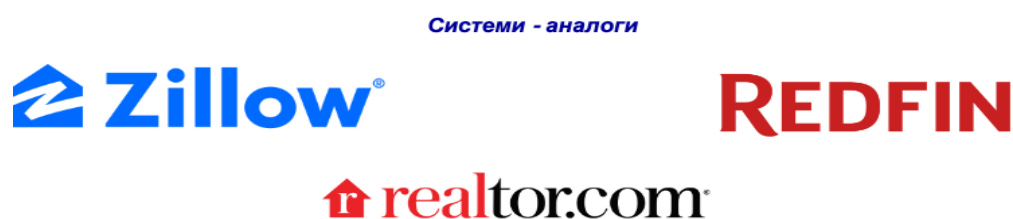


Рисунок 1.3 — Три найпопулярніші сервіси-оцінювачі нерухомості

Найпопулярніша система для оцінки нерухомості в Сполучених Штатах — Zillow [12]. Zillow може оцінити будь-який тип об'єкта нерухомості, використовуючи для цього власну запатентовану автоматизовану модель оцінки нерухомості, яка називається Zestimate [13]. Zestimate має дуже низький середній рівень помилок, приблизно 5% . Цей алгоритм оцінки вартості об'єктів нерухомості враховує різні фактори, такі як фізичні та економічні фактори об'єкту нерухомості, а також фактори державного регулювання. За допомогою великої кількості даних (на даний час близько 10,5 мільйонів статистичних даних - ця база даних з часом збільшується), включаючи дані про продажі нерухомості, подібні нерухомості в тому ж районі та довідкові дані, Zillow обчислює оцінку вартості нерухомості. Окрім цього, система надає цікаву функцію, яка відрізняє її від інших двох систем-аналогів, — створення персоналізованого облікового запису користувачів з усіма дотичними функціями (збереження розглянутих та цікавих для користувача об'єктів нерухомості, інформування про появу схожих за критеріями до збережених об'єктів, і т.д.).

Наступним розглянемо Redfin [14]. Це брокерська компанія з нерухомості, вона може зібрати необхідну інформацію (обробляються такі ж групи факторів, що і для Zillow) про будь-яку нерухомість із надійних джерел, і основна її мета це забезпечити перехід потенційних клієнтів від роздумів про продаж будинку до процесу його продажу за допомогою одного із своїх досвідчених експертів. Його джерела є надійними та свій автоматизований алгоритм оцінки Redfin Estimate забезпечують досить низький середній рівень помилок, приблизно 8%. Якщо користувач задоволений оцінкою, він може продовжити процес продажу, поговоривши з одним із експертів Redfin, який надасть безкоштовний порівняльний аналіз ринку для більш точної оцінки житла, дана функція відрізняє даний сервіс від інших.

І останній — Realtor.com [15]. Ця автоматизована система оцінки нерухомості, на відміну від розглянутих систем-аналогів, не надає у вільному доступі інформацію про середній відсоток помилок. Проте відомо, що дана система реалізує оцінку квартир (інші типи нерухомості не підтримуються) за допомогою власного алгоритму оцінки — Realtor Estimate, який спираючись на даних про недавно продані схожі об'єкти та даних з різних сторонніх ресурсів, що описують всі ті фактори, що і проаналізовані вище системи) розраховує оцінку.

В результаті проведеного аналізу була створена таблиця 1.1, яка структурує дані аналізу за критеріями, що є важливими для нас.

Таблиця 1.1 — Порівняльний аналіз існуючих рішень

	Zillow	Redfin	Realtor
Автоматизована оцінка нерухомості	+	+	+
Середній відсоток помилок	~ 5 %	~ 8 %	Не відомо
Підтримуваний тип оцінюваних об'єктів	Всі типи	Всі типи	Квартири

Продовження таблиці 1.1

	Zillow	Redfin	Realtor
Автоматизований алгоритм оцінки	Zestimate	Redfin Estimate	Realtor Estimate
Використання додатково порівняльного підходу	+	—	+
Врахування різних груп факторів (окрім факторів ризику та часу)	+	+	+
Персоналізований обліковий запис користувача	+	—	—
Врахування фактору часу	—	—	—
Врахування фактору ризику	—	—	—

Дані системи є дійсно корисними та багатофункціональними, проте важливих для інвесторів засобів, а саме врахування факторів часу та ризику, що є важливим для прогнозування та забезпечення достовірності результатів, немає. Тому створення системи, яка буде не тільки розраховувати оцінку нерухомості, але також враховувати час та ризик, буде корисним та доцільним.

1.7 Формулювання вимог до системи для оцінки нерухомості

Проаналізувавши найбільш відомі сервіси-аналоги, а також враховуючи основні етапи автоматизованої моделі оцінки, можна сформулювати основні вимоги до розроблюваного ресурсу.

Щодо функціональних вимог, то система має:

- надавати користувачу засоби для вводу даних про об'єкт нерухомості, а також надавати мапу розташування об'єктів для зручності;

- надавати експертам засоби для аналізу введених користувачем даних та для додавання додаткової необхідної інформації;
- мати можливість витягувати необхідну інформацію із зовнішніх ресурсів;
- робити базову перевірку на валідність введених користувачем даних;
- виконувати попередню підготовку даних для автоматизованої моделі оцінки;
- розраховувати оцінку ризиків;
- розраховувати оціночну вартість нерухомості ґрунтуючись на даних ринку, попиту та інших наданих факторів, але без врахування фактору часу;
- надавати можливість робити прогнозування оцінки в потрібний користувачеві час;
- формувати звіти про оцінку нерухомості, та віддавати його користувачу(шляхом інформування поштою та/або відображення в графічному інтерфейсі);
- формувати та надавати історію заявок користувача на обрахунок оцінки нерухомості;
- інформувати користувача про зміну оцінки його нерухомості;
- містити інформацію про стан ринку нерухомості в США, таку як середню ціну на житло, тенденції змін цін, попит та пропозицію тощо (це допоможе користувачам зрозуміти, як ринок впливає на вартість їх нерухомості);
- мати інструменти аналізу ринку нерухомості, такі як графіки та діаграми, які демонструють зміну цін на нерухомість;
- надавати засоби для створення власного профілю (засоби для авторизації та автентифікації);
- мати можливість редагування профілю користувача;
- мати можливість відновлення паролю, в разі його втрати.

Окрім функціональних вимог, варто також описати нефункціональні вимоги, такі як вимоги до інтерфейсу, апаратні та програмні, а також операційні вимоги.

Вимоги до інтерфейсу:

- адаптивність інтерфейсу;
- інтуїтивно-зрозумілий інтерфейс: забезпечення користувача на високому рівні вказівками стосовно його можливих дій, а також генерувати належний зворотний зв'язок на його запити;
- забезпечення простоти переходу від виконання однієї функції до іншої;
- реакція системи на всі типи запитів повинна бути однозначною і зрозумілою і, по можливості, простою;
- максимальна простота використання і готовність в повній мірі задовільнити запити користувача при розв'язанні визначеного класу задач;
- розумний та простий дизайн: дизайн повинен бути чистим, простим та розумним, щоб не заважати користувачу знайти потрібну інформацію;
- забезпечення швидкості роботи, щоб користувач міг швидко отримувати необхідну інформацію.

Апаратні та програмні вимоги: коректне відображення сайту у Chrome, Safari, Microsoft Edge. А також в операційних системах Linux, MacOS, Windows.

Щодо операційних вимог, то система повинна забезпечувати:

- Безпеку та конфіденційність;
- Надійність веб-застосунку;
- Коректність виконання задачі оцінки;
- Керування помилками.

Важливо визначити не тільки вимоги до системи, але також потоки інформації і матеріальних об'єктів. Все це описується контекстною діаграмою, що зображена на рисунку 1.4.



Рисунок 1.4 — Контекстна діаграма системи

Таким чином з рисунка 1.4 видно, що на вхід даної системи приходять необхідна інформація про об'єкт нерухомості, а результатом роботи даної системи буде спрогнозована оцінка цього об'єкта. База правил та обмежень, згідно з якою виконується робота системи, — нормативно-правова база оцінки нерухомості у США, а механізмами роботи системи виступають користувач(тобто потенційний інвестор) та експерт нерухомості.

1.8 Висновок

Даний розділ розпочався оглядом теми інвестицій в нерухомість, що наштовхнуло на ідею створити допоміжний ресурс, який на основі якісних та кількісних ознак нерухомості зможе спрогнозувати оцінку нерухомості, враховуючи фактори часу та ризику. Було наведено обґрунтування актуальності

даної розробки та розглянуто майбутній продукт як з точки зору майбутнього користувача, так і з точки зору розробників.

Для повного розуміння предметної області було проведено аналіз нормативно-правової бази оцінки нерухомості, існуючих методик для визначення оцінки нерухомості, а також факторів, які впливають на оцінку. Спираючись на згаданому аналізі, було прийнято рішення щодо використання наукової методології оцінки (AVMs) та побудовано узагальнену технологію оцінки для цього методу.

На завершення даного розділу, було проведено порівняльний аналіз існуючих систем-аналогів, для виявлення як сильних так і слабких сторін, а також для впевненості в потребі даної розробки. Спираючись на останньому, було наведено узагальнений опис задачі та визначено основні вимоги до розроблюваного застосунку. Окрім вимог, було визначено потоки інформації і матеріальних об'єктів та представлено їх за допомогою контекстної діаграми системи.

РОЗДІЛ 2. ПРОЕКТНІ РІШЕННЯ ДЛЯ СИСТЕМИ ВИЗНАЧЕННЯ ОЦІНОЧНОЇ ВАРТОСТІ НЕРУХОМОСТІ

2.1 Проектні вирішення задачі оцінки нерухомості

Як вже згадувалося раніше, у даній роботі пропонується використання автоматизованих моделей оцінки (AVMs). Однак, існує кілька методів автоматизованих моделей оцінки, які можуть вирішити задачу оцінки нерухомості, основні з них зображено на рисунку 2.1.



Рисунок 2.1 — Основні способи AVMs для оцінки нерухомості

Так, спираючись на рисунок 2.1, виділяють регресійний аналіз, алгоритми машинного навчання та методи багатокритеріальної оптимізації. Далі пропонується детальніший огляд кожного з них.

2.1.1 Регресійний аналіз для оцінки нерухомості

Регресійний аналіз — це статистичний метод, який показує зв'язок між двома чи більше змінними. Зазвичай виражається у вигляді графіка, метод перевіряє зв'язок між залежною змінною та незалежними змінними. Як правило, незалежна змінна (змінні) змінюється разом із залежною (змінними) і регресійний аналіз намагається відповісти, які фактори мають найбільше значення для цієї зміни [16].

Регресійний аналіз є одним із методів оцінки нерухомості, який базується на статистичних моделях для аналізу взаємозв'язку між різними факторами та їх

впливом на вартість нерухомості. Основною функцією регресійного аналізу є визначення математичної формули, яка описує залежність між вхідними даними та ціною нерухомості. Крім того, регресійний аналіз може бути використаний для прогнозування майбутньої вартості нерухомості на основі раніше зібраних даних та тенденцій ринку. Однак, даний підхід не працюватиме з нечіткими факторами та є досить обмеженим у врахуванні складних зв'язків між ними.

2.1.2 Машинне навчання для оцінки нерухомості

Застосування алгоритмів машинного навчання: цей спосіб використовує комп'ютерні алгоритми для аналізу великої кількості даних про ринок нерухомості, такі як ціни на схожі об'єкти, географічні особливості, стан нерухомості та інші фактори. Алгоритми машинного навчання можуть навчитися розпізнавати та враховувати ці фактори для автоматичного визначення вартості нерухомості.

Алгоритми машинного навчання можуть бути використані для оцінки нерухомості в США. Зазвичай це використовується в тому випадку, якщо необхідно оцінити велику кількість нерухомості, наприклад, у випадку оцінки портфеля нерухомості, де є багато об'єктів для оцінки.

Для застосування алгоритмів машинного навчання для оцінки нерухомості, необхідно зібрати велику кількість даних про нерухомість. Після попередньої обробки можна перейти до застосування алгоритмів машинного навчання, які можуть бути використані для побудови моделі оцінки нерухомості. Для цього можуть бути використані різні типи алгоритмів машинного навчання, такі як дерева рішень, нейронні мережі та інші. Після навчання моделі на підготовлених даних, вона може бути використана для прогнозування вартості нерухомості на основі нових даних про нерухомість.

2.1.3 Багатокритеріальна оптимізація для оцінки нерухомості

Згідно з Міжнародними стандартами оцінки, ринкова вартість визначається як розрахункова величина, рівна грошовій сумі, за яку майно повинно переходити з рук в руки на дату оцінки між добровільним покупцем і добровільним продавцем в результаті комерційної угоди після адекватного маркетингу, при цьому покладається, щоб кожна зі сторін діяла компетентно, виважено і без примусу [17]. Тобто поняття ринкової вартості "прив'язана" до колективного сприйняття і поведінки учасників ринку. Воно визнає різні фактори, які можуть впливати на транзакції на ринку, і відокремлює їх від інших притаманних для нього або неринкових міркувань, що впливають на вартість.

У практичній діяльності часто зустрічаються схожі задачі, які полягають в пошуку кращого (оптимального) рішення при наявності різних несвідомих один до одного критеріїв оптимальності. Наприклад, прийняття рішення про будівництво дороги в об'їзд міста має враховувати такі фактори, як виграш міста в цілому по міркуваннях екології, програш окремих підприємств і фірм, наприклад, через зменшення проїжджаючих через місто потенційних покупців і багато інших. Якщо такого роду завдання вирішуються методами математичного програмування, то говорять про завдання багатокритеріальної оптимізації [18].

Завдання багатокритеріальної оптимізації виникають у тих випадках, коли є кілька цілей, які не можуть бути відображені одним критерієм (наприклад, вартість і надійність). Потрібно знайти точку області допустимих рішень, яка мінімізує або максимізує всі такі критерії. Саме аналіз деяких факторів об'єктів нерухомості, як кількісних так і якісних, розгортає перед нами задачу багатокритеріальної оптимізації. А проектування цієї вартості не тільки в поточний момент, а і в деякий момент в майбутньому, відкриває перед нами й задачу прогнозування. Більше того, врахування фактору ризику, відриває перед нами й інші задачі, де потрібно залучати штучний інтелект.

2.1.4 Аналіз основних методів багатокритеріальної оптимізації

Багатокритеріальні задачі не мають однозначного рішення. Це обумовлює множинність способів знаходження єдиного рішення[19] (рисунок 2.2).

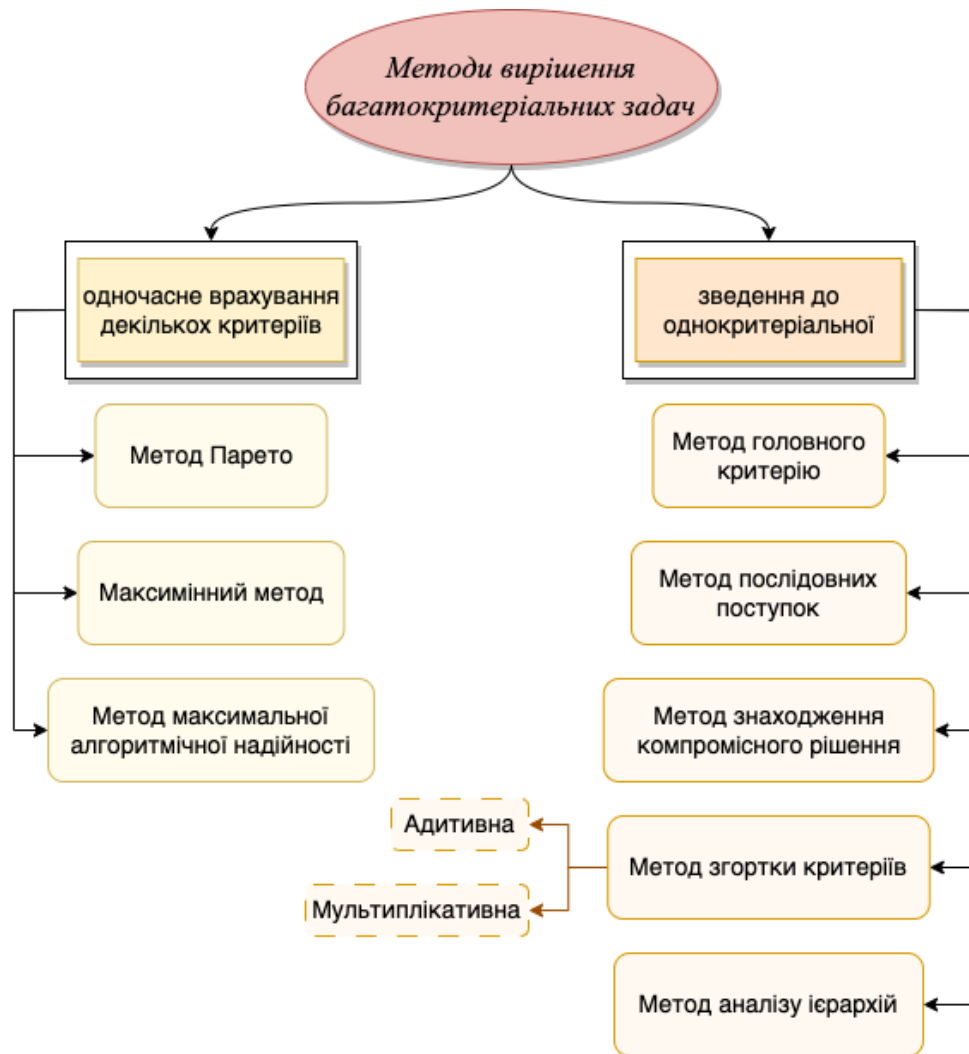


Рисунок 2.2 — Основні методи вирішення багатокритеріальної задачі

Один із основних підходів до вирішення багатокритеріальних задач — це одночасне врахування декількох критеріїв.

Найбільш відомим з цих методів є метод Парето. Даний метод полягає у відмові від визначення однієї «найкращої» альтернативи. Визначається множина так званих «рівно ефективних» рішень, які кращі за інші (що не належать до цієї множини) хоча б за одним критерієм, та принаймні не гірші за рештою критеріїв.

Іншим методом даного підходу є максимінний підхід. Прийняття рішень за допомогою методу максимінної стратегії ґрунтується на припущенні, що кількість критеріїв J , щодо яких приймається рішення, і число можливих варіантів вирішення V обмежені і заздалегідь відомі. У цьому випадку повна умова задачі задається матрицею (таблиця 2.1).

Таблиця 2.1 — Початкові дані максимінної стратегії прийняття рішень

V_j / J_i	V_1	V_2	...	V_n
J_1	a_{11}	a_{12}	...	a_{1n}
...
J_m	a_{m1}	a_{m2}	...	a_{mn}
\min_i	a_{x1}	a_{j1}	...	a_{hn}

Аналізуючи табличні дані по стовпцях, вибирається для кожного з них мінімальний елемент, тобто найбільш критичний показник-критерій. Після попереднього вибору мінімального елемента, в цьому рядку відбираємо максимальне значення, тобто варіант з найбільшим значенням критерію. Отримане рішення володіє гарантованими властивостями: максимізація найгіршого показника (2.1):

$$V_{opt} = \max(\min(a_{ij})) \quad (2.1)$$

Ключові значення параметрів алгоритму — коефіцієнти a_{ij} визначаються на основі аналізу апріорних даних про досліджувану ситуації, якщо вона має чітку і відому структуру, статистично — в іншому випадку.

Прийняття рішень за допомогою композиції нечітких множин узагальнює цей метод, знімаючи обмеження на наявність чіткої і відомої структури витрат і прибутків від реалізації можливих варіантів, а також на наявність репрезентативної інформації про функціонування досліджуваного об'єкта. Невідомі коефіцієнти розглядаються в цьому випадку як функції належності нечітких множин. Таким чином, використання теорії нечітких множин є для

цілого класу вирішуваних в економіці завдань найбільш кращим, а в ряді випадків і єдино можливим.

Ще одним методом є метод максимальної алгоритмічної надійності. У даному методі замість вимоги максимізації критеріїв пропонується ввести деякі чисельні значення \bar{J}_i заданих критеріїв J_i , які допоможуть вирішувати задачу, яка для кожного конкретного i визначає множину допустимих стратегій S_i , кожна з яких обмежує множину допустимих стратегій управління за відповідним критерієм. Перетин цих областей визначає "трубку рішень" — множину стратегій S_0 , допустимих за всіма критеріями. В якості основного критерію J_0 на S_0 пропонується критерій максимальної алгоритмічної надійності, суть якого полягає в тому, що вибирається та стратегія, яка при можливих прогнозованих відхиленнях ходу технологічного процесу з мінімальною ймовірністю призведе до порушення умови задачі. Цьому критерію, очевидно, задовольняє вісь "трубки рішень" $S_{\text{опт}}$.

Проаналізувавши методи даного підходу, бачимо, що жодний з методів не підходить конкретно під нашу задачу, адже деякі методи або просто не вирішують поставленої задачі, або є надто складними в реалізації, що не є доцільним. Тому було прийнято рішення розглянути інший підхід вирішення багатокритеріальних задач, а саме зведення багатокритеріальної задачі до однокритеріальної за допомогою введення узагальненого показника, який так чи інакше об'єднує в собі окремі критерії. Такий показник називають показником корисності або суперкритерієм. Як правило, суперкритерій не має фізичної інтерпретації, він являє собою безрозмірну величину, яка дозволяє порівняти альтернативи. Більше значення суперкритерію відповідає кращій альтернативі.

Один із методів такого підходу, є метод головного критерію. Даний метод пропонує оптимізацію за одним найбільш важливим критерієм(2.2), всі інші критерії розглядаються як додаткові обмеження.

$$J_1(x) \rightarrow \max, x \in G, \quad (2.2)$$

де $J_2(x) \geq C_2, J_3(x) \geq C_3, \dots, J_m(x) \geq C_m,$

C_2, \dots, C_m — ті значення показників, які повинні бути обов'язково досягнуті, щоб забезпечити нормальне функціонування системи.

Частіше зустрічається, що локальні критерії об'єктивно рівнозначні, або що ЛПР (іноді об'єктивно, а іноді і в силу некомпетентності) не можна визначити головний критерій. У цих випадках метод головного критерію не забезпечить вірний результат, і варто звернутись до методів, які ґрунтуються на упорядкуванні заданої множини критеріїв та послідовній оптимізації по кожному з них. Таким методом є метод послідовних поступок. Цей метод, також як і метод головного критерію, застосовується у випадку коли критерії впорядковані за важливістю. Опишемо його для задачі виду (2.3):

$$J_i(x) \rightarrow \min, \quad i \in I, x \in G, J_1(x) \geq J_2(x), \dots, J_m(x) \quad (2.3)$$

Сутність методу послідовних поступок полягає в тому, що вихідна багатокритеріальна задача замінюється послідовністю однокритеріальних задач, область допустимих рішень яких звужується від задачі до задачі за допомогою додаткових обмежень, які враховують вимоги до критеріїв. При формулюванні кожної задачі, по відношенню до важливішого критерію робиться поступка, величина якої залежить від вимог задачі і оптимального рішення за цим критерієм.

Наступні два методи, пропонують вирішення задачі за допомогою зведення множини критеріїв до одного шляхом введення експертних вагових коефіцієнтів для кожного з критеріїв, і саме такі методи чудово підходять під нашу задачу:

1. Метод згортки (угруповання) критеріїв (лінійна, мультиплікативна, інші згортки).
2. Метод аналізу ієрархій.

Щодо першого методу згортки критеріїв, то розрізняють адитивну і мультиплікативну згортки. Найпростішою з них є лінійна, адитивна згортка (2.4). Вона визначається наступним виразом:

$$F(J_k) = \sum_{k=1}^m \lambda_k \frac{J_k(x)}{\max J_k(x)} \rightarrow \max, \quad \text{де } \lambda_k > 0; \sum_{k=1}^m \lambda_k = 1, \quad (2.4)$$

де λ_k — вагові коефіцієнти, важливості локальних критеріїв. Вага критеріїв визначається експертним шляхом, при цьому більш важливому критерію надається більша вага. Сума значень ваги всіх критеріїв повинна дорівнювати одиниці. Якщо всі локальні критерії мають однакову важливість, то їх можна винести за знак суми і скоротити, так як вони не змінюють точки максимуму.

Знаменник являє собою оцінку альтернативи вирішення J_k за оптимальним k -им локальним критерієм при повному відкиданні інших. У цьому випадку всі складові будуть не більше одиниці і безрозмірними величинами. Це важливо, оскільки локальні критерії оптимальності можуть мати різну розмірність (наприклад, грошові одиниці, тимчасові одиниці, відсотки). Таким чином, даний метод досить трудомісткий, але він найбільш гнучкий, оскільки дозволяє варіювати важливістю кожного локального критерію.

Мультиплікативна згортка оперує результатом множення степенів локальних критеріїв. Мультиплікативні згортки легко зводяться до адитивних шляхом логарифмування результату множення локальних критеріїв.

Вибір виду згортки — це доволі складне завдання, потребує хорошого розуміння сутності досліджуваного процесу. Якщо ЛПР схильний до застосування локальних критеріїв, і спирається на союз "або" (якщо або перший критерій виконується, або другий), то потрібно використовувати адитивну згортку критеріїв. Якщо ж сполучником є "і" (якщо і перший критерій виконується, і другий), то необхідно використовувати мультиплікативну.

Однак, даний метод може бути менш ефективним для визначення оцінки нерухомості, адже задача визначення оцінки ОН є доволі складною з багатьма критеріями, тож потребує декомпозиції на менші підзадачі та ієрархічне визначення відносної важливості критеріїв. Метод АІ дозволяє більш детально проаналізувати структуру задачі та відносну важливість кожного критерію. Він також дозволяє врахувати взаємозв'язки між критеріями та їх вплив на рішення.

2.1.5 Метод аналізу ієрархій для оцінки нерухомості

Метод аналізу ієрархій (МАІ) [20] — математичний інструмент системного підходу до вирішення складних проблем прийняття рішень. Основне застосування методу полягає в підтримці прийняття рішень за допомогою ієрархічної декомпозиції завдання. Цей метод розроблений американським математиком Томасом Сааті, який написав про нього книги, розроблював програмні продукти і протягом 20 років проводив симпозиуми ISAHN [21] (англ. International Symposium on Analytic Hierarchy Process). Розроблений ним стандартний алгоритм МАІ має п'ять основних етапів (рисунок 2.3).

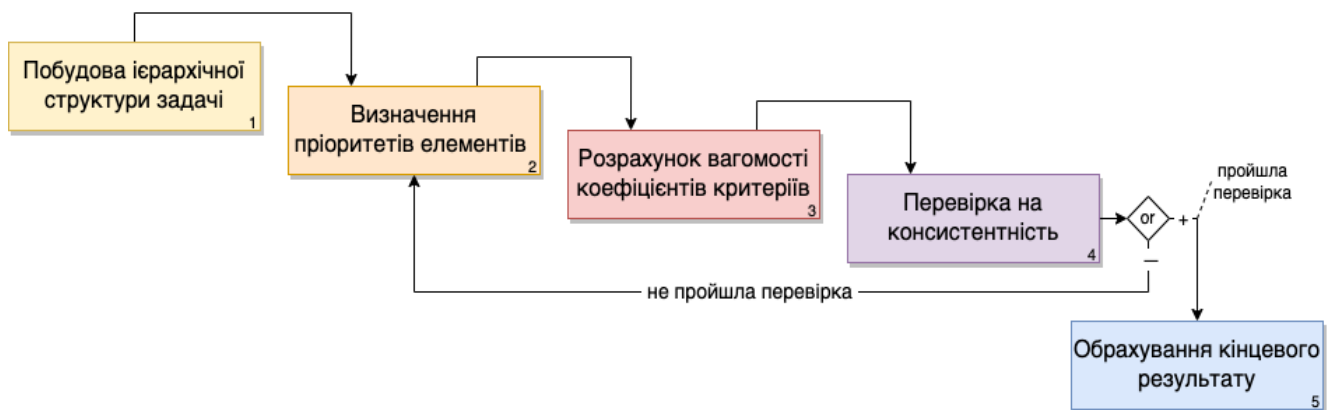


Рисунок 2.3 — Основні етапи МАІ

Розглянемо кожен з представлених етапів методу аналізу ієрархій детальніше. Однак, при цьому ми будемо інтегрувати їх до вирішення задачі обчислення оцінки нерухомості, враховуючи особливості саме нашої задачі. Це призведе до внесення деяких змін до стандартного алгоритму для забезпечення більш точних результатів.

Етап 1. Побудова ієрархічної структури задачі.

Аналіз проблеми прийняття рішень в МАІ починається з побудови ієрархічної структури, яка включає мету, критерії й інші фактори, що впливають на вибір. Кожен елемент ієрархії може представляти різні аспекти задачі, що розв'язується, причому до уваги можуть бути прийняті як матеріальні, так і

нематеріальні чинники, вимірювані кількісні параметри та якісні характеристики, об'єктивні дані і суб'єктивні експертні оцінки. В нашому випадку критеріями виступатимуть фактори, що описані в розділі 1.3.

Етап 2. Визначення пріоритетів елементів ієрархічної структури.

Наступним етапом аналізу є визначення пріоритетів, що представляють відносну важливість або перевагу елементів побудованої ієрархічної структури, за допомогою процедури попарних порівнянь. Для встановлення відносної важливості елементів ієрархії використовується спеціальна шкала (рисунок 2.4).

Рівень переваги	Визначення	Пояснення
1	Відсутність переваги	Внесок альтернатив до цілі однаковий
2	Слабка перевага	
3	Посередня перевага	Досвід та судження трохи сприяють одній з альтернатив над іншою
4	Більш ніж посередня перевага	
5	Сильна перевага	Досвід та судження сильно сприяють одній з альтернатив над іншою
6	Більш ніж сильна перевага	
7	Дуже сильна або продемонстрована перевага	Перевага дуже сприятлива до однієї з альтернатив, її домінування продемонстровано на практиці
8	Дуже, дуже сильна перевага	
9	Екстремальна перевага	Докази, що сприяють одній з альтернатив над іншою є найвищим можливим порядком підтвердження
1,1 - 1,9	Значення, близькі до відсутності переваги	Коли альтернативи дуже близькі додавання знаків після коми дозволяє показати наявність різниці

Рисунок 2.4 — Шкала відношень Сааті [22]

Ця шкала Сааті надає експерту можливість ставити у відповідність ступеням переваги одного фактору перед іншим — деяке число.

Етап 3. Розрахунок вагомості коефіцієнтів критеріїв.

Ранжування елементів, що аналізуються з використанням матриці попарних порівнянь, здійснюється на підставі аналізу головних власних векторів матриці попарних порівнянь. Власний вектор забезпечує упорядкування пріоритетів, а власне значення є мірою узгодженості (однорідності) суджень. Обчислення головного власного вектору W додатної квадратної матриці A проводиться на підставі рівності(2.5):

$$AW = \lambda_{max}W \quad (2.5)$$

де λ_{max} — максимальне власне число матриці A ,

$$W = \begin{pmatrix} W_1 \\ W_2 \\ \dots \\ W_n \end{pmatrix} \text{ — головний власний вектор матриці } A.$$

Власний вектор W нормований (сума його елементів дорівнює 1), тому він є вектором пріоритетів для матриці A . Якщо власний вектор W ненормований, то потрібно його пронормувати, для цього необхідно знайти суму всіх його елементів і створити новий вектор W' елементи якого є відношенням елементів вектору W на знайдену суму.

На практиці обчислення власного вектору виконуються до досягнення заданої точності(2.6). За достатню для практики точність можна взяти 0.01 незалежно від порядку матриці.

$$\xi : e^T |W^k - W^{k-1}| \leq \xi \quad (2.6)$$

Етап 4. Перевірка на консистентність.

На практиці внаслідок помилок експертів, обмеженості простору можливих значень переваг при попарних порівняннях можливе недотримання консистентності, того, наскільки послідовним та узгодженим було прийняття рішень у процесі аналізу ієрархій.

Однорідність (узгодженість) суджень експертів оцінюється індексом однорідності (T) (2.7):

$$T = \frac{|\lambda_{max} - n|}{n-1}, \quad (2.7)$$

Відношення узгодженості обчислюється як відношення(2.8) обчисленого індексу узгодженості T до табличного індексу $M(T)$ з таблицки 2.2.

$$U = \frac{T}{M(T)}, \quad (2.8)$$

Таблиця 2.2 — Табличні індекси

Розмір матриці (n)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Табличний індекс (μ_T)	0	0	0,52	0,89	1,11	1,25	1,35	1,40	1,45	1,49	1,52	1,54	1,56	1,58	1,59

Матриця попарних порівнянь вважається узгодженою, якщо $\frac{T}{M(T)} < 0.1$. Якщо матриця попарних порівнянь не є узгодженою, то необхідно переглянути порівняння, і перебудувати матрицю попарних порівнянь.

Етап 5. Обрахування кінцевого результату.

Отримання кінцевої оцінки для значень введеної квартири(тобто альтернативи, в даному випадку вона одна).

Процес прогнозування ґрунтуватиметься на описаному способі розрахунку оцінки за допомогою методу аналізу ієрархій. Однак, тут додатково потрібно робити коригування ваг на часовому проміжку. Причини для зміни важливості факторів можуть бути різні, наприклад, забудова в екологічно чистому районі, зміна тенденцій для вибору нерухомості та інші. Все це впливає на оцінку, все це є ризики. Управління ризиками в оцінці об'єктів нерухомості є важливим для забезпечення ефективного та безпечного процесу інвестування в нерухомість. Класифікація можливих ризиків інвестування у нерухомість, їх оцінка та процес управління над ними може бути реалізований як з використання стандартних методів, таких як аналіз фундаментальних показників, аналіз трендів ринку, оцінка стабільності економіки тощо [23] так і з використанням алгоритмів машинного навчання, таких як регресійна модель, модель на основі дерева рішень, нейронні мережі чи навіть MAI [24 - 25].

2.2 Функціональний аналіз системи оцінки нерухомості

При проектуванні інформаційної системи одним з найважливіших кроків є проведення функціонального аналізу. Це означає, що потрібно врахувати всі сформовані вимоги до системи (описані в розділі 1.7), а також особливості реалізації MAI для задачі оцінки нерухомості, які були розглянуті в

попередньому пункті. На основі цього можна визначити, які функції повинна забезпечувати система та які задачі вона повинна виконувати. Результатом функціонального аналізу є функціональна модель системи (рисунок 2.5).

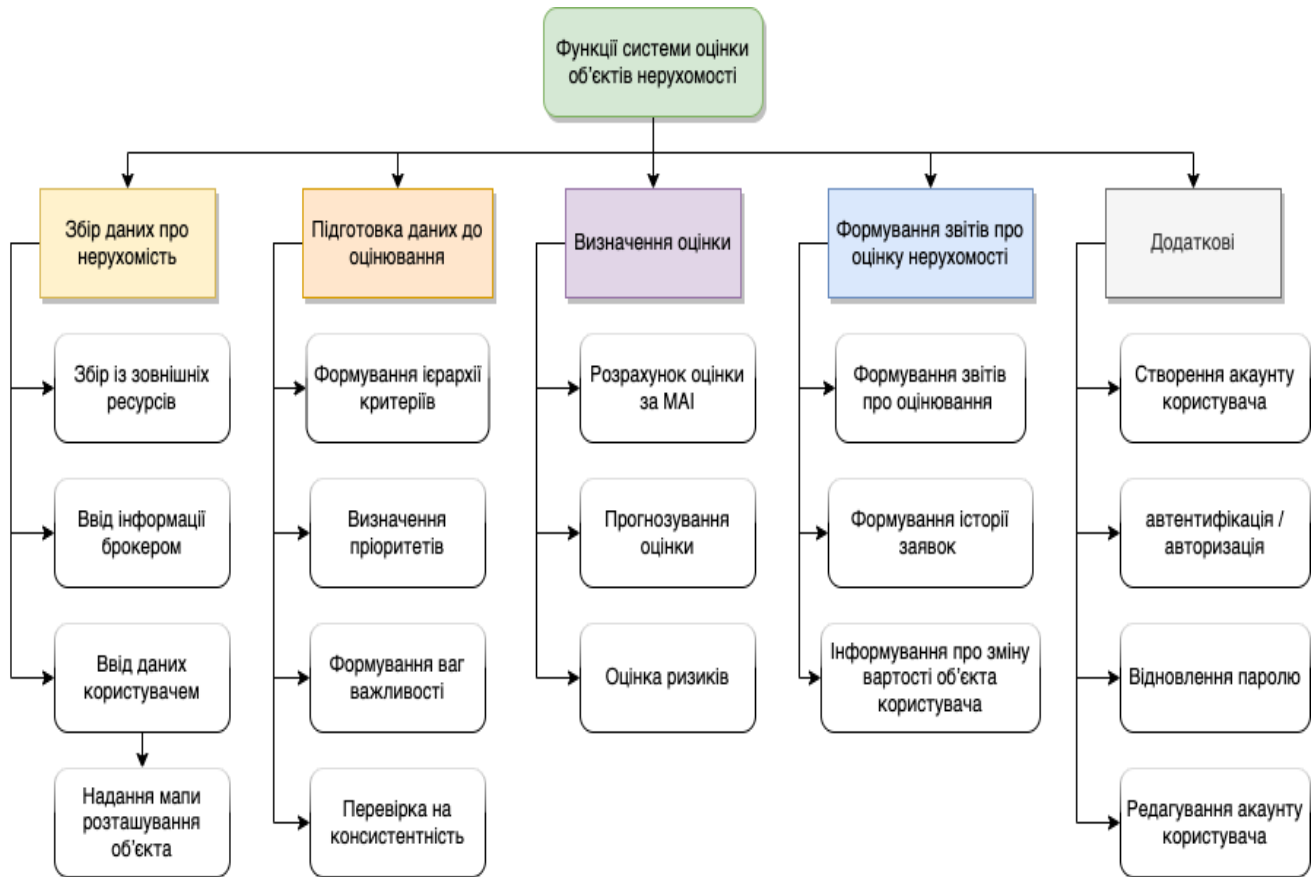


Рисунок 2.5 — Функції системи оцінки об'єктів нерухомості

Модель з рисунку 2.5 є основою для подальшого проектування, розробки та тестування інформаційної системи, оскільки дозволяє зрозуміти, як система повинна працювати та які функції повинні бути реалізовані.

Після проведення початкового функціонального аналізу і отримання функціональної моделі системи можна перейти до створення діаграми прецедентів (рисунок 2.6).

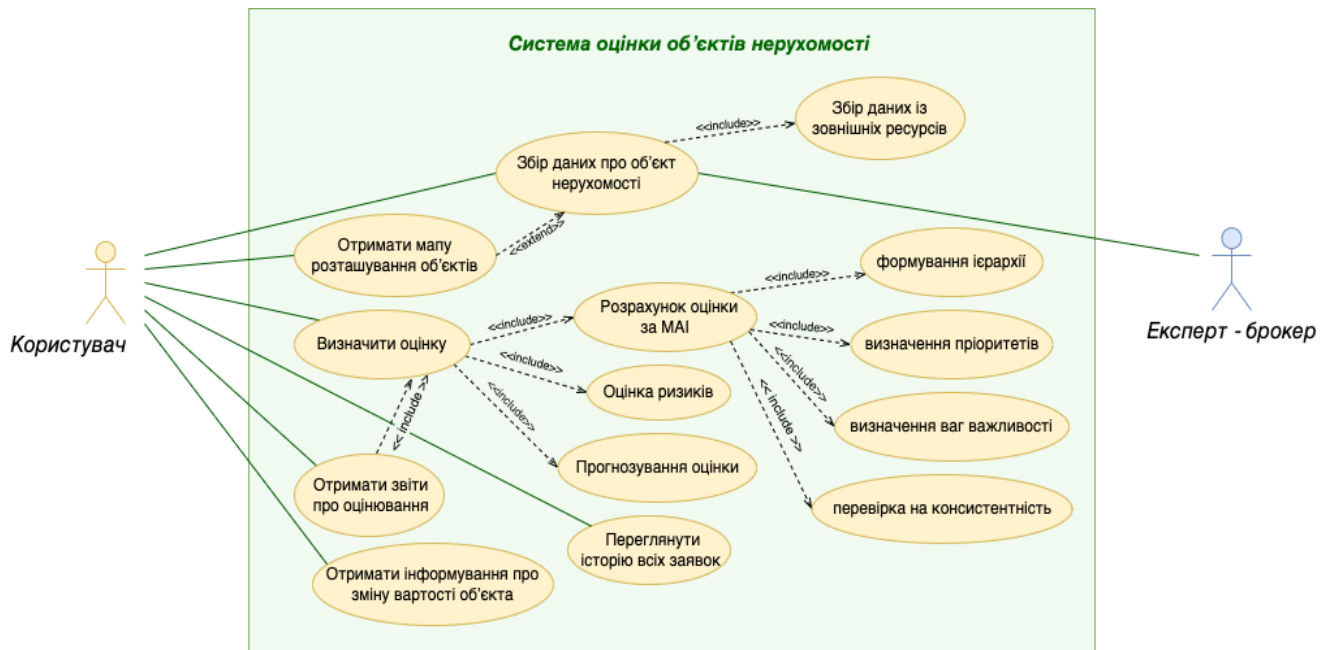


Рисунок 2.6 — Діаграма використання

Представлена діаграма відображає взаємодію користувачів з системою, що безпосередньо стосується процесу оцінки об'єкта нерухомості.

2.3 Архітектура системи для визначення оцінки нерухомості

Реалізація системи для оцінки нерухомості, описаної в даній роботі, буде представлена у вигляді веб-застосунку. Архітектуру веб-застосунку (рисунок 2.7) спроектовано на основі багатoshарової моделі, що складається з трьох основних компонентів: клієнтської сторони, серверної сторони та бази даних:

- Клієнтська сторона — це фронтенд-складова веб-застосунку, що відповідає за інтерфейс користувача.
- Серверна сторона — це бекенд-складова веб-застосунку, що відповідає за логіку програми, обробку запитів користувача та взаємодію з базою даних.
- База даних (БД) — це складова системи управління даними, що використовується для зберігання, організації та доступу до даних.

Окрім основних компонентів, архітектура також містить механізми авторизації та аутентифікації, систему кешування даних, а також сторонні API.

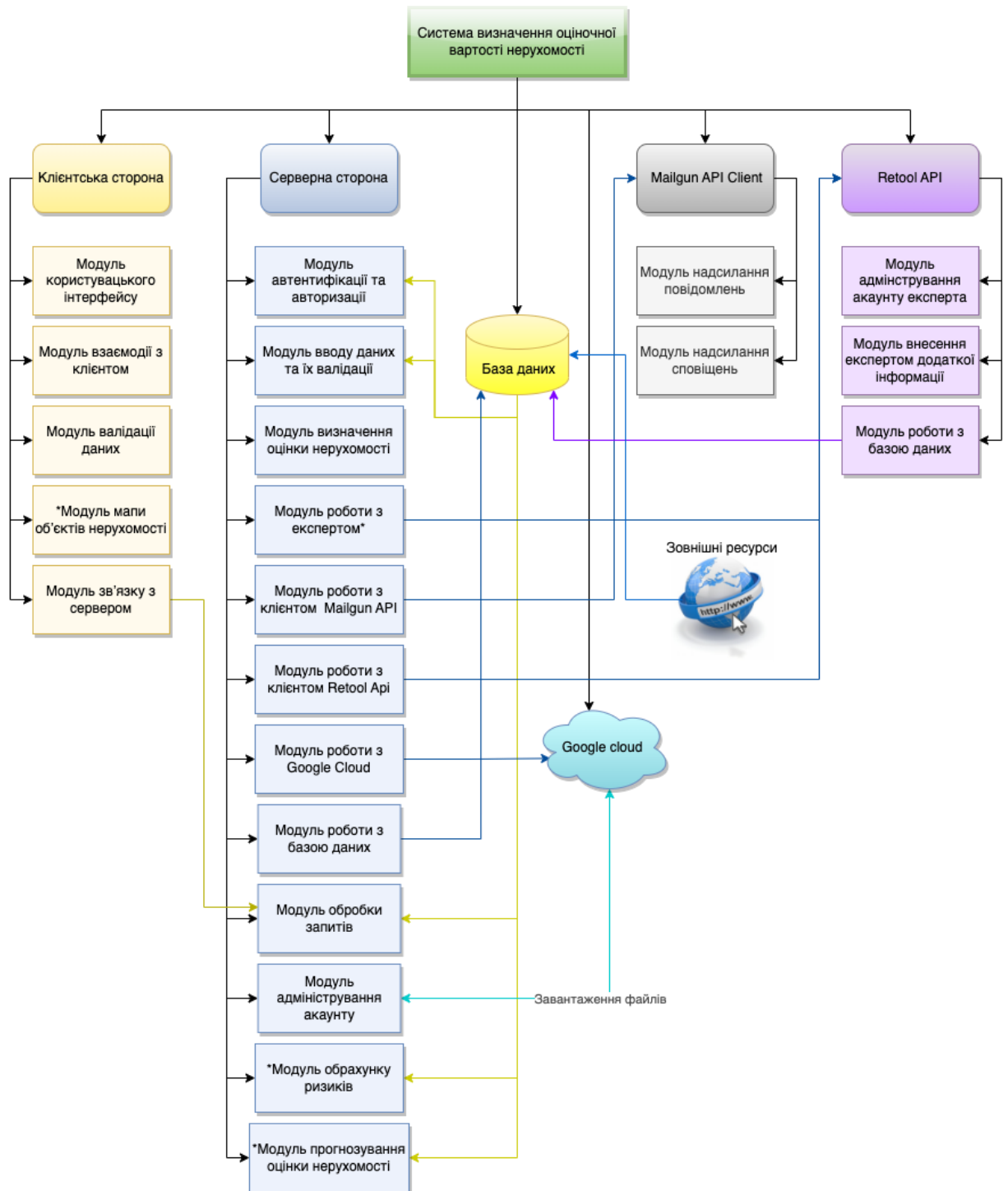


Рисунок 2.7 — Архітектура системи для розрахунку оцінки нерухомості

На рисунку 2.7 зображені сторонні API, такі як Mailgun API для надсилання електронних листів, Google Cloud для роботи з документами та

забезпечення їх безпеки, та ReTool API для комунікації з експертом нерухомості. Крім цього, рисунок демонструє не тільки компоненти архітектури, але й їх модулі; також зображує зв'язки між всіма складовими архітектури.

2.3.1 Модуль визначення оцінки нерухомості

Для подальшої структуризації, потрібно розглянути головний процес системи — процес оцінки об'єкта нерухомості. Слід відзначити, що в поточній версії продукту не буде реалізовано модулі, які забезпечують збір інформації про об'єкти нерухомості з зовнішніх ресурсів, так як цей процес є дуже ресурсозатратним. Тому, на даному етапі не буде реалізації залежних від цього модулів (на рисунку 2.7 вони представлені із «*»). Внаслідок цього, система не зможе проставити пріоритети критеріїв автоматично, тому цю функцію виконуватимуть експерти нерухомості. Враховуючи всі ці аспекти, процес оцінки нерухомості, що буде реалізовано в даній версії продукту, можна представити діаграмою у нотатції BPMN (рисунку 2.8).

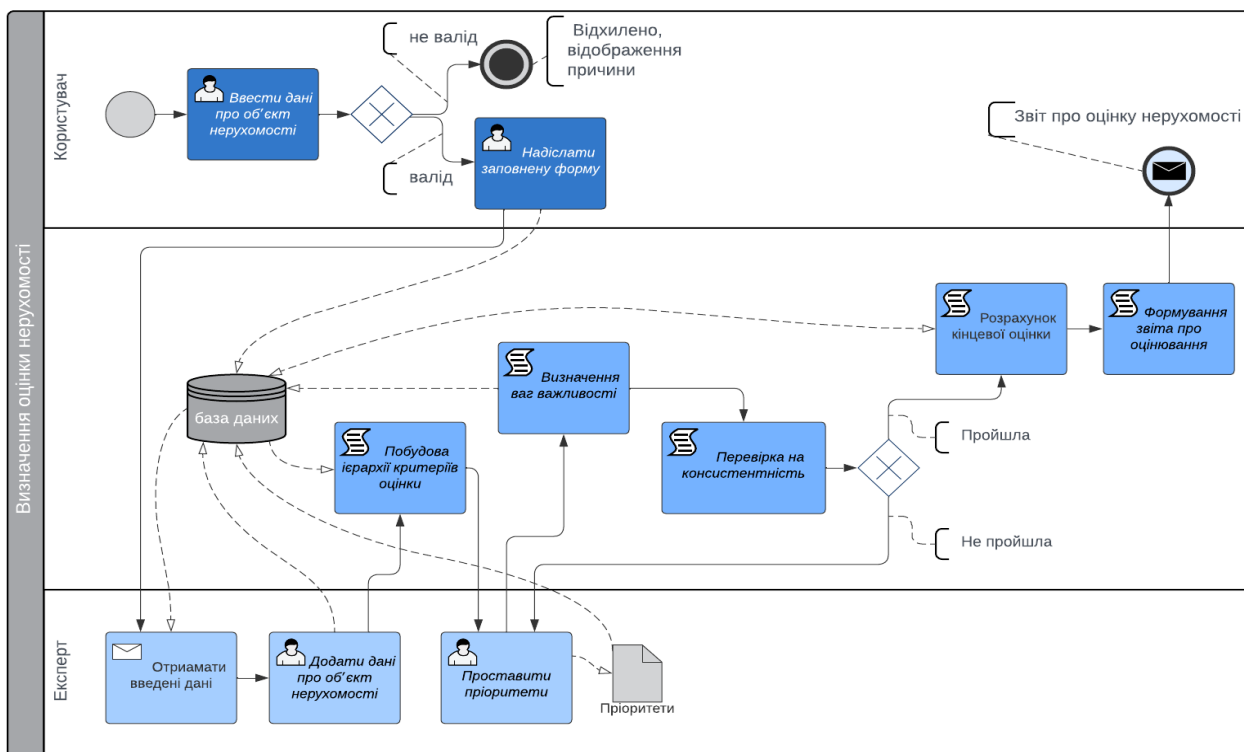


Рисунок 2.8 — Процес визначення оцінки нерухомості

Діаграма з рисунку 2.8 структурувала бізнес-процес визначення оцінки нерухомості та описала послідовність дій, що виконуються в рамках даного процесу, а також зв'язки між різними елементами процесу.

2.4 Проектування бази даних для розроблюваної інформаційної системи

На архітектурі інформаційної системи (рисунок 2.7) компонент «База Даних» вимагає детальнішого проектування. Проектування БД — це процес створення структури БД, яка відображає сутності та їх взаємозв'язки, що повинні зберігатися в БД, та визначення способів доступу до цих даних.

На даному етапі роботи, після проведення детального функціонального аналізу, визначення потреб користувача та вимог до системи, уже можна спроектувати загальну архітектуру БД. Так, у нас повинен бути користувач, який може мати багато об'єктів нерухомості для обрахунку. Кожен об'єкт містить багато критеріїв, а кожен критерій має свою вагу, яка є унікальною для кожного запиту. Отже, будемо мати наступні сутності: «Користувач», «Об'єкт нерухомості», «Запит на розрахунок», «Критерій», «Історія запиту».

Зв'язки між сутностями, їх тип та пояснення до них (таблиця 2.3) — є важливим етапом для побудови моделей баз даних.

Таблиця 2.3 — Зв'язки між сутностями

№	Сутності, що утворюють зв'язок	Тип зв'язку	Пояснення
1	користувач — об'єкт нерухомості	1...1 : 0...N	Один користувач може мати багато ОН. Один ОН задається тільки одним користувачем.
2	об'єкт нерухомості — критерій	1... N : 1... N	Один критерій описує багато ОН. Один ОН має містити декілька критеріїв(атрибут зв'язку «значення критерію»).

Продовження таблиці 2.3

№	Сутності, що утворюють зв'язок	Тип зв'язку	Пояснення
3	запит на розрахунок — історія запиту	1...1 : 1...N	Один запит на розрахунок має декілька записів в історії. Один запис історії відноситься до одного запиту.
4	ОН — запит на розрахунок	1...1 : 1...N	Один ОН може мати декілька запитів на розрахунок. Один запит відноситься до одного ОН.
5	критерій — запит на розрахунок	1...N : 1...N	До одного запиту може відноситись декілька критеріїв. Один критерій може оброблятися декількома запитами (атрибут зв'язку «вага»).

Розуміючи зв'язки між сутностями та їх тип можна побудувати концептуальну модель бази даних (рисунок 2.9), яка описує сутності та їх взаємозв'язки у системі. Вона є абстрактною та не залежить від конкретної реалізації системи або БД.

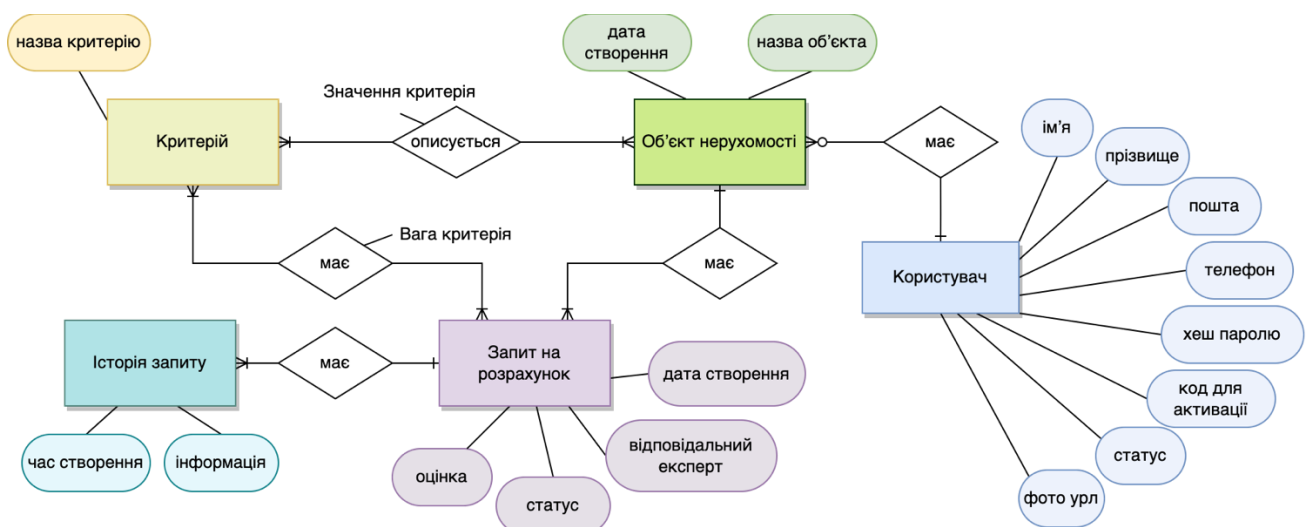


Рисунок 2.9 — Концептуальна модель бази даних

На основі концептуальної моделі бази даних можна розробити логічну модель бази даних, яка буде більш деталізованою та конкретною. Логічна модель БД (рисунок 2.10) включає в себе таблиці з полями та зв'язками між ними.

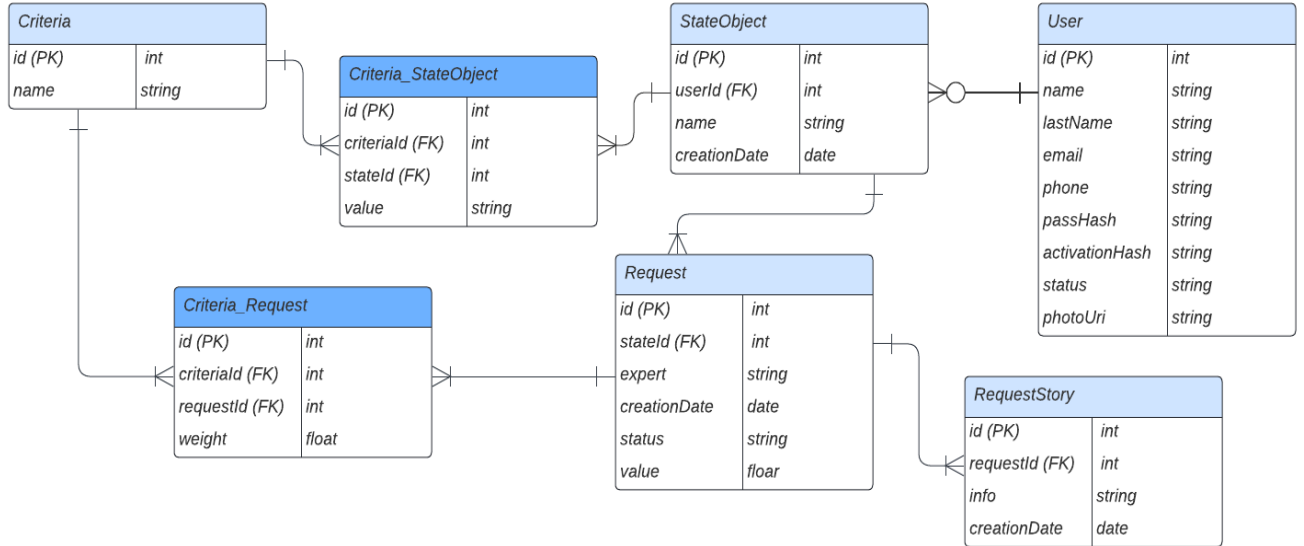


Рисунок 2.10 — Дата-логічна модель бази даних

Фізична модель бази даних побудована згідно вимог та правил PostgreSQL (рисунок 2.11) описує спосіб зберігання даних в конкретній системі керування базами даних. Вона включає в себе таблиці, їхні поля та взаємозв'язки між ними.

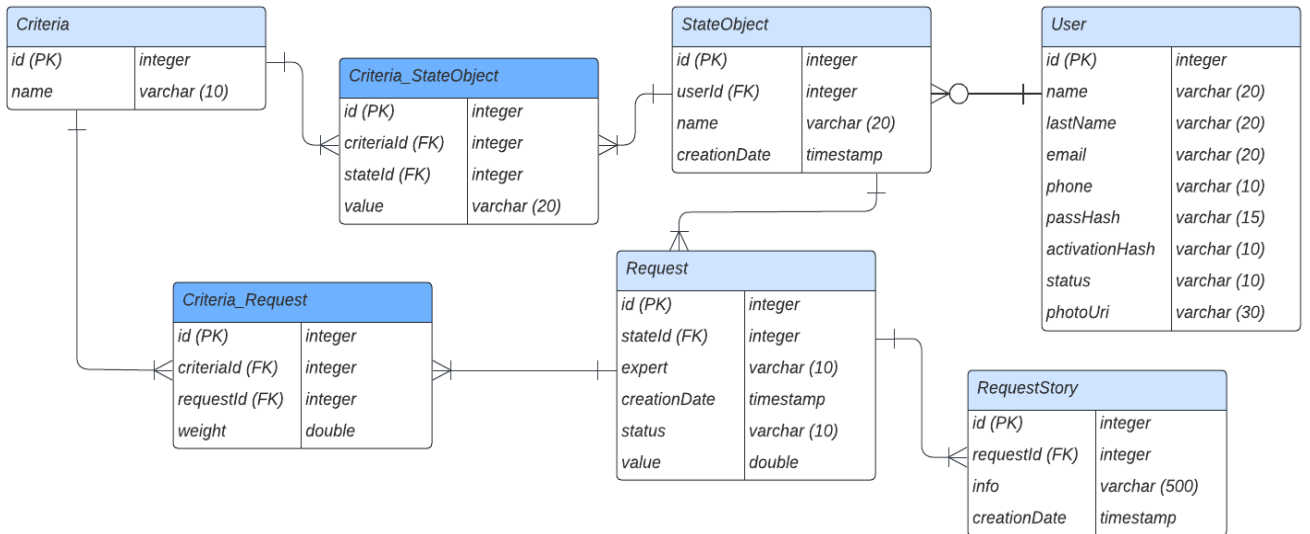


Рисунок 2.11 — Фізична модель бази даних

В результаті проектування БД було представлено архітектуру бази даних, яка повинна забезпечити коректну реалізацію необхідних функцій для даної версії продукту. Більше того, така архітектура уже створює власний набір даних для реалізації функції прогнозування, яка планується бути розробленою в наступних версіях застосунку.

2.5 Висновок

Отже, в даному розділі було описано та проаналізовано проектні вирішення задачі обчислення оцінки об'єкта нерухомості. На основі цього було прийнято рішення використовувати для поставленої задачі метод аналізу ієрархій. Даний вибір зумовлено тим, що метод аналізу ієрархій дозволяє зрозумілим і раціональним чином структурувати складну проблему прийняття рішень у вигляді ієрархії, порівняти і виконати кількісну оцінку альтернативних варіантів рішення. Метод АІ дозволяє більш детально проаналізувати структуру задачі та відносну важливість кожного критерію, а також дозволяє врахувати взаємозв'язки між критеріями та їх вплив на рішення.

Окрім згаданого, було проведено функціональний аналіз, де було визначено не тільки функції, але і варіанти використання системи. Для загальної повноти процесу проектування інформаційної системи, було наведено архітектуру програмного продукту, де було описано основні компоненти, їх модулі та зв'язки між ними. Більше того, було проведено декомпозицію найважливішого процесу системи — визначення оцінки нерухомості, дану декомпозицію було представлено за допомогою діаграми у нотації BPMN.

На завершення даного розділу, було спроектовано архітектуру бази даних, яка відображає сутності та їх взаємозв'язки, що повинні зберігатися в базі даних, та визначення способів доступу до цих даних. Для структуризації цього етапу, було побудовано концептуальну, логічну та фізичну модель бази даних.

РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ВИЗНАЧЕННЯ ОЦІНОЧНОЇ ВАРТОСТІ НЕРУХОМОСТІ

3.1 Обґрунтування вибору програмних засобів

З поміж великої кількості програмних засобів, які можуть бути використані для реалізації веб-сайту, в даній роботі були обрані та використані наступні: мови програмування TypeScript та JavaScript, мову розмітки HTML та мову стилів CSS; Node.js — платформу для виконання JavaScript на стороні сервера, яка дозволяє створювати швидкі та ефективні веб-додатки та API; фреймворк Express.js для створення маршрутів та обробки запитів та фреймворк для реалізації клієнтського інтерфейсу — Next.js; об'єктно-реляційна система керування базами даних — PostgreSQL та Sequelize як ORM, яка дозволяє працювати з БД; key-value базу даних Redis та бібліотеку BullMQ, що використовуються для роботи з чергами повідомлень; платформу Docker для розгортання, управління та запуску контейнеризованих додатків; програму ngrok для роботи з тунелюванням, яка дозволяє легко створювати публічний URL для доступу до веб-додатка на локальному комп'ютері; середовище розробки — PhpStorm. Далше детальніше буде обґрунтовано чим зумовлено такий вибір.

Вибір мов програмування TypeScript та JavaScript, а також платформи Node.js, яка робить JavaScript(TypeScript) мовою загального значення, може бути обґрунтовано наступними причинами:

- Широке використання, яке забезпечує багато ресурсів та бібліотек, що дозволяють зручно і ефективно розробляти веб-додатки.
- Сумісність з браузерами: JavaScript є мовою, яка виконується безпосередньо в браузері користувача, тому вона дозволяє створювати динамічні інтерфейси та взаємодіяти з користувачем.
- TypeScript, який є суперсетом JavaScript, дозволяє використовувати типи даних, що підвищує стабільність і безпеку програмного коду.
- Більш масштабована альтернатива Apache: Node.js може обробляти кілька одночасних підключень.

- Парадигма «JavaScript everywhere». Тобто не потрібно перемикаати мови програмування між серверною та зовнішньою розробкою, що призводить до меншої кількості файлів та коду.

Щодо HTML та CSS, то вони є стандартними мовами розмітки та оформлення веб-сторінок і їх використання не потребує додаткового обґрунтування .

Вибір фреймворків Express.js та Next.js для розробки веб-додатків на Node.js зумовлені наступними причинами:

- Швидкість та продуктивність. Express.js — це мінімалістичний фреймворк, який забезпечує швидкий та ефективний маршрутизатор HTTP-запитів. Next.js, з іншого боку, забезпечує швидку передачу контенту на стороні клієнта та на стороні сервера, що дозволяє зменшити час завантаження сторінки та збільшити продуктивність.
- Простота використання. Обидва фреймворки мають простий та зрозумілий інтерфейс, який дозволяє швидко створювати веб-додатки та API-інтерфейси. Express.js дозволяє легко маршрутизувати HTTP-запити, встановлювати middleware та здійснювати обробку помилок. Next.js надає можливість створювати статичні та динамічні сторінки з використанням повноцінного рендерингу на стороні сервера (SSR) та статичної генерації сторінок (SSG).
- Велика спільнота, яка забезпечує активну підтримку, розвиток та допомогу у вирішенні проблем.
- Розширюваність: Express.js та Next.js можна легко розширювати за допомогою сторонніх пакетів та бібліотек.

Вибір PostgreSQL можна обґрунтувати тим, що ця СУБД може запускати динамічні веб-сайти та веб-програми як опцію стека LAMP, а також однією з основних причин є її висока надійність, що робить цю СУБД ідеальною для розробки веб-сайтів зі складними вимогами до даних. Для ефективного управління базою даних використовується ORM Sequelize, яка є дуже гнучкою та підтримує парадигму «JavaScript everywhere».

Вибір Redis та BullMQ для роботи з чергами повідомлень може бути обґрунтовано декількома причинами. По-перше, Redis є швидкою та ефективною key-value базою даних, яка може зберігати значення ключів в пам'яті, що дозволяє швидко отримувати доступ до даних. Це робить Redis ідеальним вибором для роботи з великим обсягом даних, що потребують швидкого доступу. По-друге, BullMQ є потужною бібліотекою для роботи з чергами повідомлень у Node.js, що дозволяє легко створювати та управляти завданнями, що повинні бути виконані асинхронно. Використання BullMQ дозволяє досить просто та ефективно реалізувати функціонал чергової обробки даних на веб-сайті. Таким чином, використання Redis та BullMQ може значно спростити та прискорити роботу з даними на веб-сайті, зменшивши витрати часу та ресурсів на обробку даних.

Docker — це програмне забезпечення з відкритим кодом, найпопулярніша платформа для управління контейнерами. Він вирішує безліч завдань, пов'язаних зі створенням контейнерів, розміщенням в них додатків, управлінням процесами, а також тестуванням ПЗ і його окремих компонентів. Він потрібен для більш ефективного використання системи і ресурсів, швидкого розгортання готових програмних продуктів, а також для їх масштабування і перенесення в інші середовища з гарантованим збереженням стабільної роботи.

Основні переваги ngrok — простота використання, можливість швидко створити тунель для локального веб-сервера та можливість доступу до нього з будь-якої точки Інтернету, без необхідності використання зовнішніх хмарних платформ.

Середовище розробки PhpStorm — це комерційне інтегроване середовище розробки для різних мов програмування (PHP, JS, TS та ін.) від компанії JetBrains. Дане середовище ідеально підходить для розробки веб-додатків. Є дуже багато переваг даного середовища, та з основних це підвищення продуктивності, яке досягається за допомогою автодоповнення коду, та можливість роботи з базою даних в самому середовищі розробки.

3.2 Структура програмного забезпечення(ПЗ)

Структура програмного забезпечення включає в себе різноманітні складові, які можна розділити на три основні рівні: рівень додатку (application level), рівень середовища виконання (runtime environment level) та рівень операційної системи (operating system level) [33].

Щодо рівня операційної системи, то веб-сайт можна запускати на будь-якій операційній системі, що підтримує Node.js: Windows, Linux або macOS. Рівень додатку включає в себе програмний код, що написаний для виконання затребуваної функціональності системи, бази даних, бібліотеки та інші компоненти, що використовуються в програмі. Отже, узагальнена структура розробленої програми подана на рисунку 3.1. Більше того, даний рисунок зображує файли конфігурації Docker, що належить до середовища виконання та дозволяє запускати веб-сайти та додатки у контейнерах.

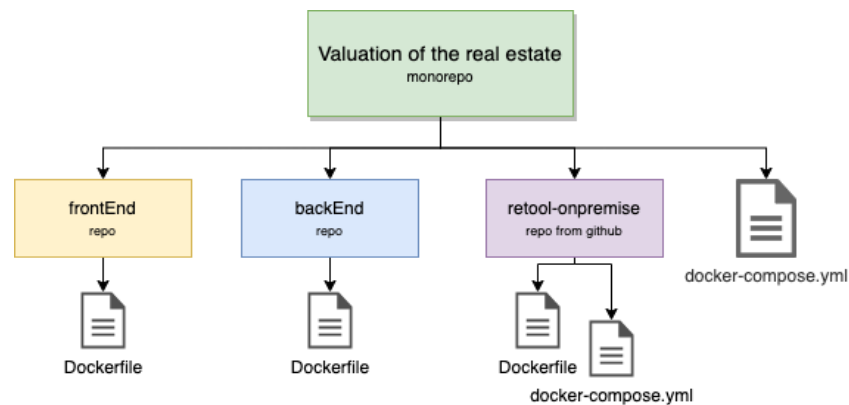


Рисунок 3.1 — Узагальнена структура програми

ReTool On-Premise — це продукт, що надає можливість запускати середовище ReTool на власному сервері, що забезпечує збереження даних та конфіденційності. Репозиторій ReTool On-Premise містить всі необхідні складові для встановлення та налаштування цього продукту на власному сервері.

3.2.1 Структура програмного забезпечення серверної сторони

Структура програмного забезпечення серверної сторони визначається компонентами, які відповідають за різні аспекти роботи сервера. Основними компонентами серверної сторони для проекту на Node.js з використанням фреймворку Express є веб-сервер, маршрутизація, middleware для обробки запитів, база даних, системи аутентифікації та авторизації користувачів, а також різноманітні сервіси для роботи з іншими системами. Ці компоненти дозволяють створити функціональну та безпечну систему для обробки запитів від клієнтів.

Розпочнемо з огляду підсистеми «backEnd», його структурна схема представлена на рисунку 3.2.

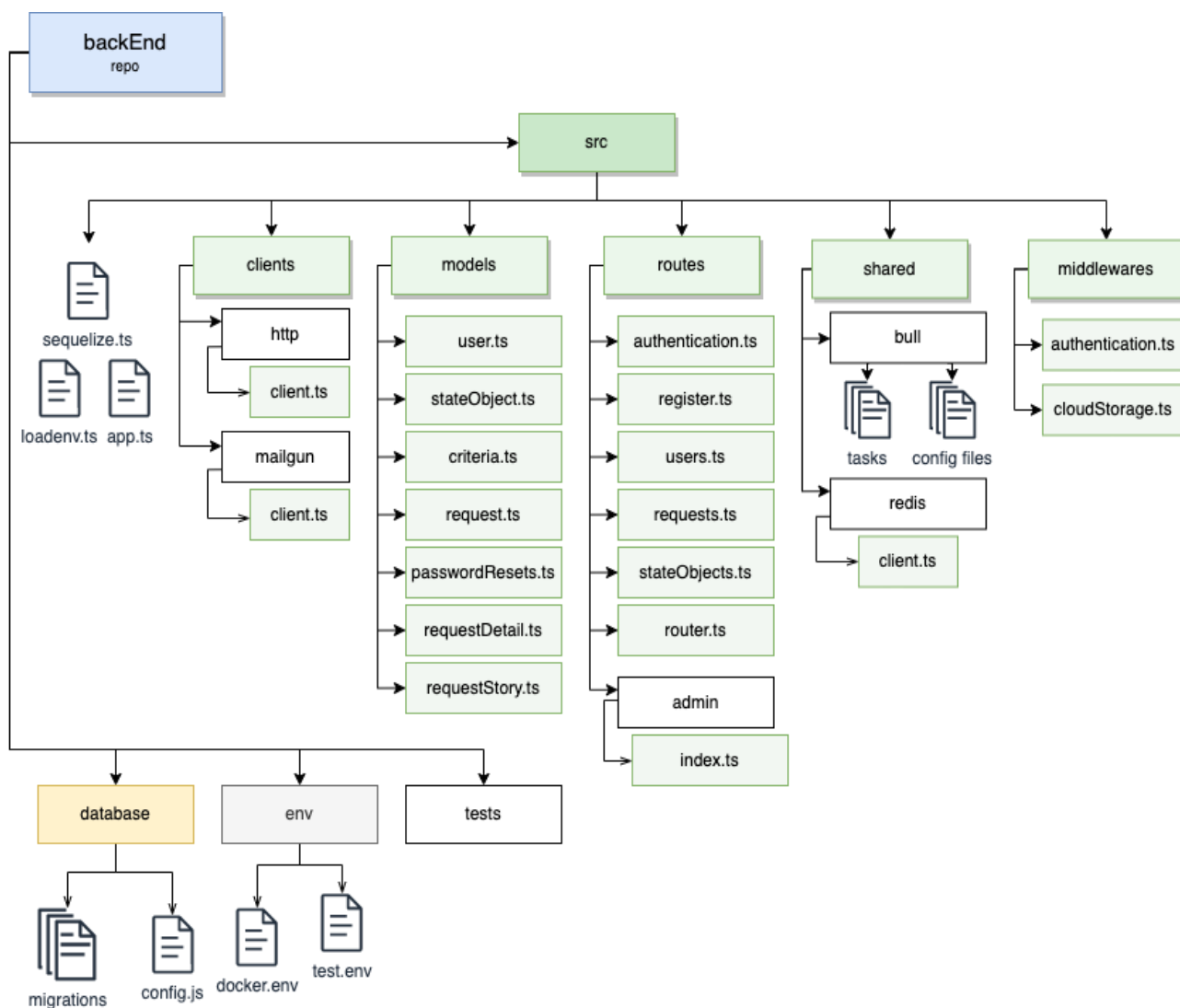


Рисунок 3.2 — Структурна схема ПЗ підсистеми «backEnd»

З рисунку 3.2 видно, що структурна схема підсистеми «backEnd» складається з наступного:

- репозиторій `src`, який містить файл з точкою входу (entry point) — `app.ts`, а також інші конфігураційні файли. Окрім цього, у цій папці є і інші підпапки, такі як `clients`, `models`, `middleware`, `routes`, `shared`, `utils`, які організовують код за певною логікою і допомагають зберігати його структурованим і легким для розуміння; файли з цих підпапок містять реалізацію різних функцій і маршрутів (routes) серверної частини додатку;
- репозиторій `database`, в якому описано файл конфігурації бази даних — `config.js`, та всі файли міграцій;
- репозиторій `env`, що містить файли з конфігураційними даними для проекту(на даний момент є окремо для Docker та окремо для тестів);
- репозиторій `tests`, який окрім тестів, містить ще і конфігураційні файли для них.

Розглянемо детальніше основні програмні модулі репозиторію `src` та структуруємо це наступною таблицею 3.1.

Таблиця 3.1 — Специфікація програмних модулів підсистеми «backEnd»

Модуль	Пояснення
`app.ts`	Відповідає за конфігурацію та налаштування сервера. Цей файл є точкою входу.
`sequelize.ts`	Відповідає за налаштування з'єднання з базою даних та ініціалізацію ORM Sequelize.
`loadenv.ts`	Відповідає за завантаження конфігураційних змінних з файлу <code>.env</code> .
`models`	
`user.ts`	Відповідає за опис моделі «користувача» в базі даних.
`criteria.ts`	Відповідає за опис моделі «критерій» в базі даних.

Продовження таблиці 3.1

<code>`stateObject.ts`</code>	Відповідає за опис моделі «об'єкт нерухомості» в базі даних.
<code>`request.ts`</code>	Відповідає за опис моделі «запит на розрахунок» в базі даних.
<code>`passwordResets.ts`</code>	Відповідає за опис моделі «запит на зміну пароллю» в базі даних.
<code>`requestStory.ts`</code>	Відповідає за опис моделі «історія запиту» в базі даних.
<code>`requestDetails.ts`</code>	Відповідає за опис моделі «деталі запиту на розрахунок» в базі даних.
<code>` clients `</code>	
<code>http : `client.ts`</code>	Відповідає за налаштування HTTP клієнта в проєкті, цей файл експортує клас <code>`HttpClient`</code> , який містить методи для відправки HTTP запитів на сервер. Цей клас призначений для реалізації зв'язку між клієнтом та сервером.
<code>mailgun : `client.ts`</code>	Відповідає за створення клієнта для взаємодії з API Mailgun. В цьому файлі виконується налаштування конфігурації клієнта, створення і налаштування HTTP-запитів до API Mailgun для відправлення та отримання повідомлень електронної пошти.
<code>` routes `</code>	
<code>`authentication.ts`</code>	Відповідає за маршрути для аутентифікації та авторизації.
<code>`register.ts`</code>	Відповідає за маршрут для реєстрації нового користувача.
<code>`users.ts`</code>	Відповідає за обробку HTTP-запитів, що пов'язані з ресурсом користувачів (запити на отримання, оновлення або видалення користувача тощо).
<code>`stateObjects.ts`</code>	Відповідає за обробку маршруту, що пов'язаний з отримання об'єктів нерухомості.
<code>admin : `index.ts`</code>	Відповідає за маршрути, що пов'язані з ReTool API.

Продовження таблиці 3.1

<code>`requests.ts`</code>	Відповідає за маршрут, що пов'язаний з створенням запиту на розрахунок оцінки об'єкту нерухомості(збирання даних про об'єкт, їх валідація тощо)
<code>`router.ts`</code>	Використовується у <code>`app.ts`</code> та описує шлях до файлів обробки групи маршрутів.
<code>`middlewares`</code>	
<code>`authentication.ts`</code>	Відповідає за middleware, який виконує перевірку автентифікації користувача на основі переданого токена, з метою обмеження доступу до захищених ресурсів тільки автентифікованим користувачам.
<code>`cloudStorage.ts`</code>	Відповідає за middleware-функцію, яка відповідає за роботу сервісу хмарного сховища — Google Cloud Storage.
<code>`shared`</code>	
<code>bull</code>	Відповідає за роботу з чергами повідомлень за допомогою бібліотеки Bull.
<code>redis : `client.ts`</code>	Відповідає за створення клієнта Redis, який можна використовувати для взаємодії з Redis-сервером.

Щодо ReTool API, про який було згадано в таблиці 3.1, то ми маємо наступні файли:

- `getPendingRequests.sql` — файл, який витягує з бази даних всі об'єкти нерухомості, які потребують огляду експертом;
- `calculateWeightsAndCheckCosistency.js` — файл, який на основі розставлених експертом пріоритетів розраховує ваги важливості і перевіряє їх на консистентність;
- `finish` та `notValidInputData` — надсилають дані на обробку до запитів з нашої підсистеми «backEnd», до яких отримуємо запит за допомогою тунелювання використовуючи ngrok, який надає доступ до сайту за допомогою URL-адреси, що генерується автоматично.

3.2.2 Структура програмного забезпечення клієнтської сторони

Розглянемо підсистему «frontEnd», його структурна схема представлена на рисунку 3.3.

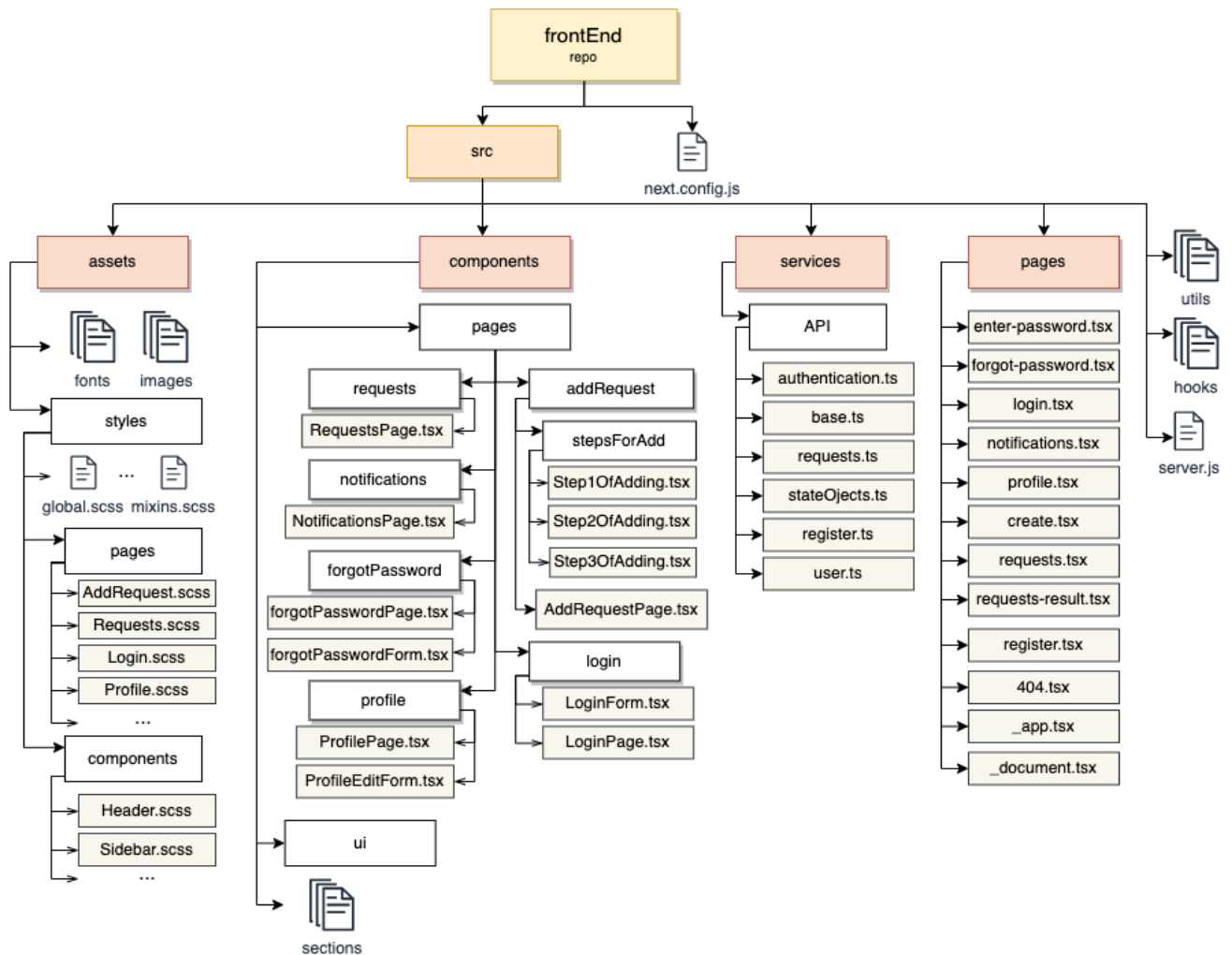


Рисунок 3.3 — Структурна схема ПЗ підсистеми «frontEnd»

З рисунку 3.3 видно, що вона складається з репозиторію `src` та файлу конфігурації — `next.config.js`. В свою чергу `src` поділяється на файл `server.js`, який призначений для кастомізації серверної логіки веб-застосунку, та наступні репозиторії: `assets`, `components`, `services`, `utils`, `pages` та `hooks`. Папка `assets` містить статичні ресурси, такі як зображення, шрифти, стилі; папка `components` містить компоненти, які можуть перевикористовуватись; папка `services` відповідає за роботу з зовнішніми API; папка `utils` використовується

для зберігання загальних функцій, які можуть бути використані в різних частинах додатку; папка `pages` містить файлову структуру для маршрутизації сторінок веб-додатку; папка `hooks` використовується для зберігання реактивних хуків (reactive hooks), які можуть бути використані в компонентах.

Розглянемо детальніше основні програмні модулі та структуруємо це наступною таблицею 3.2.

Таблиця 3.2 — Специфікація програмних модулів підсистеми «frontEnd»

Модуль	Пояснення
`assets`	
Репозиторій `pages`	Містить файли, які відповідають за стиль сторінок, які існують в системі: `AddRequest`, `Login` тощо
`AddRequest.scss`	Відповідає за стиль сторінки, яка призначена для додавання нового об'єкту нерухомості
`Login.scss`	Відповідає за стиль сторінки логіну
Репозиторій `components`	Містить файли, які відповідають за стиль основних компонентів, таких як `header`, `sidebar` тощо
`Header.scss`	Відповідає за стиль `header` (шапки) сторінок веб-сайту
...	
`components`	
Репозиторій `pages`	Містить файли реактивних компонентів, де описана логіка відображення даних, обробки подій та взаємодії з користувачем
Компонент `AddRequest`	Відповідає за «фронтенд» логіку сторінки, яка необхідна для додавання об'єкту нерухомості, має з кроки
Компонент `Notifications`	Відповідає за «фронтенд» логіку сторінки, яка необхідна для перегляду історії змін процесу обчислення оцінки заданих об'єктів нерухомості
Компонент `Requests`	Відповідає за «фронтенд» логіку сторінки, яка необхідна для перегляду всіх створених об'єктів нерухомості

Продовження таблиці 3.2

Модуль	Пояснення
Компонент `Profile`	Відповідає за «фронтенд» логіку сторінки, яка відображає профайл користувача
Компонент `ForgotPassword`	Відповідає за «фронтенд» логіку сторінки, яка необхідна при зміні паролю у випадку його втрати
Компонент `Login`	Відповідає за «фронтенд» логіку сторінки логіну
`services`	
Репозиторій `API`	Містить файли, які працюють з зовнішніми API
`authentication.ts`	Відповідає за зв'язок з API `authentication`
`register.ts`	Відповідає за зв'язок з API `register`
`user.ts`	Відповідає за зв'язок з API `users`
`request.ts`	Відповідає за зв'язок з API `requests`
`stateObjects.ts`	Відповідає за зв'язок з API `stateObjects`
`pages`	
`register.tsx`	Сторінка реєстрації в систему для користувача
`requests-result.tsx`	Сторінка після створення запиту на реєстрацію з подальшою інструкцією
`enter-password.tsx`	Сторінка створення паролю
`login.tsx`	Сторінка входу(логіну) в систему
`forgot-password`	Сторінка для відновлення паролю у випадку його втрати
`requests.tsx`	Сторінка, де відображаються всі задані об'єкти нерухомістю з детальною інформацією про них
`create.tsx`	Сторінка для додавання інформації про об'єкт нерухомості для подальшого розрахунку його оцінки
`profile.tsx`	Сторінка профайлу користувача
`notifications.tsx`	Сторінка історії заданих об'єктів нерухомості
`404.tsx`	Службова сторінка «Не Знайдено»
app.tsx, document.tsx	Відповідають за кастомізацію стартового шаблону сайту

Маючи опис усіх веб-сторінок сайту, необхідно показати як у системі відбувається перехід між ними, це можна зробити за допомогою графа переходу, який зображений на рисунку 3.4.



Рисунок 3.4 — Граф переходу між сторінками веб-сайту

Даний рисунок 3.4 структурує загальну картину веб-сайту. Кожен вузол в графі представляє сторінку веб-сайту, а ребра між вузлами представляють допустимі переходи між сторінками.

3.3 Керівництво користувачів системи визначення оцінки нерухомості

3.3.1 Керівництво користувача-клієнта

Перший тип користувачів системи оцінки ОН — це користувач-клієнт, який використовує наш сайт з метою отримати прогнозовану оцінку заданого ОН. Важливо навести інструктивне керівництво для потенційних клієнтів, які не мають досвіду роботи з даною системою, але мають намір використовувати її.

Розпочнемо з керівництва по отриманню прогнозованої оцінки заданого ОН. Після успішної автентифікації, яка включає реєстрацію та логін (дані процеси описані в додатку А), система перенаправляє користувача-клієнта на головне вікно сайту, яке містить місце для відображення всіх заданих ним ОН, а також бокове меню з переліком доступних функцій системи (рисунок 3.5).

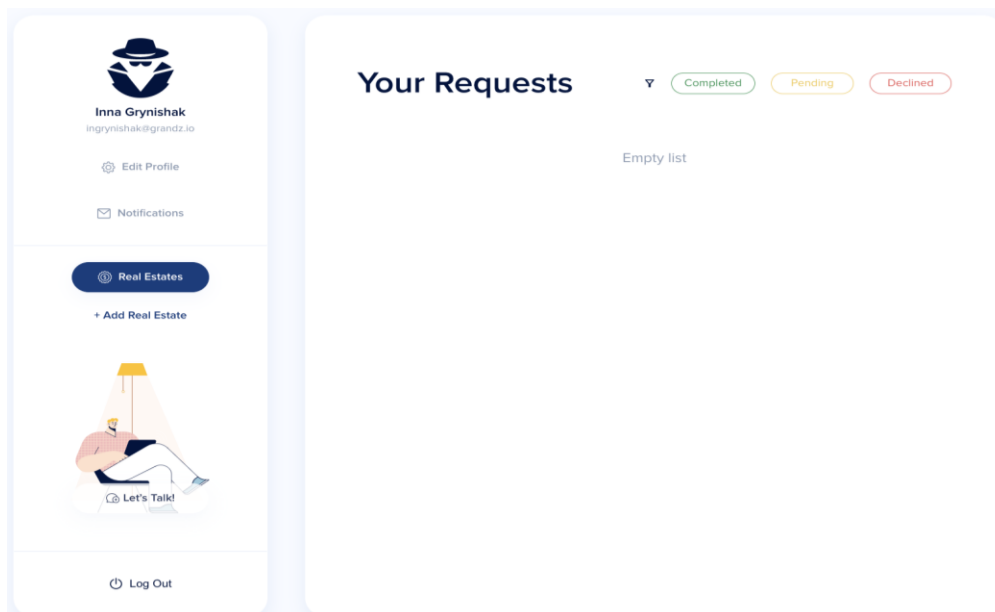


Рисунок 3.5 — Головна сторінка системи

Відповідно до процесу отримання оцінки, першим кроком є додавання інформації про ОН. Це можна зробити за допомогою підпункту бокового меню "+ Add Real Estate", що відкриє сторінку вводу даних (рисунок 3.6). Дана сторінка складається з двох послідовних форм, які необхідно заповнити коректними даними про ОН. У разі некоректного введення даних, система не дозволить відправити форму та видасть відповідні помилки (рисунок 3.7 – 3.8).

Рисунок 3.6 — Сторінка для внесення інформації про об'єкт нерухомості

Custom name of Real Estate
Test

Area in sq.m.
55

*Count of room
-6
Count can't be negative

*Street Address
Street Address
Required

City
New York

State
New York

*Longitude
Longitude
Not a number

Latitude
70.87

Next

< Back

*Floor
Floor
Not a number

Balcony

Condition of apartment
Terrible

*Condition of building
Condition of building

Required

*Date You Expect to sell or to buy
ДД.ММ.2023

Required

Add

Рисунки 3.7 – 3.8 — Приклади помилок при неправильно заповненій формі

Після успішного додавання ОН, система перенаправляє користувача на сторінку зі списком всіх доданих об'єктів з відкритою інформацією про щойно створений об'єкт та доступною фільтрацією за статусом (рисунок 3.9).

Inna Grynishak
ingrynishak@grandz.io

Edit Profile

Notifications

Real Estates

+ Add Real Estate

Let's Talk!

Log Out

Your Requests

Completed Pending Declined

Test Pending 05/07/2023 \$ 1 -

Address
New York, Wall Street, NY

Request Info

Custom Name Of Real Estate	Test
Area In Sq. M.	55 Sq. M.
Count Of Room	2
Floor	3
Balcony	Yes

Рисунок 3.9 — Сторінка всіх заданих об'єктів нерухомості

Крім вище згаданого, система автоматично створює нове повідомлення про історію об'єкту, що вказує на створення запиту на розрахунок оцінки, та повідомляє, що користувач буде проінформований електронною поштою, коли результат буде готовий. Користувач може переглянути це та інші повідомлення, які будуть створені в процесі розрахунку оцінки, натиснувши на підпункт “Notifications” у боковому меню, що відкриє сторінку з повідомленнями про зміни статусу запиту на розрахунок оцінки ОН користувача (рисунок 3.10).

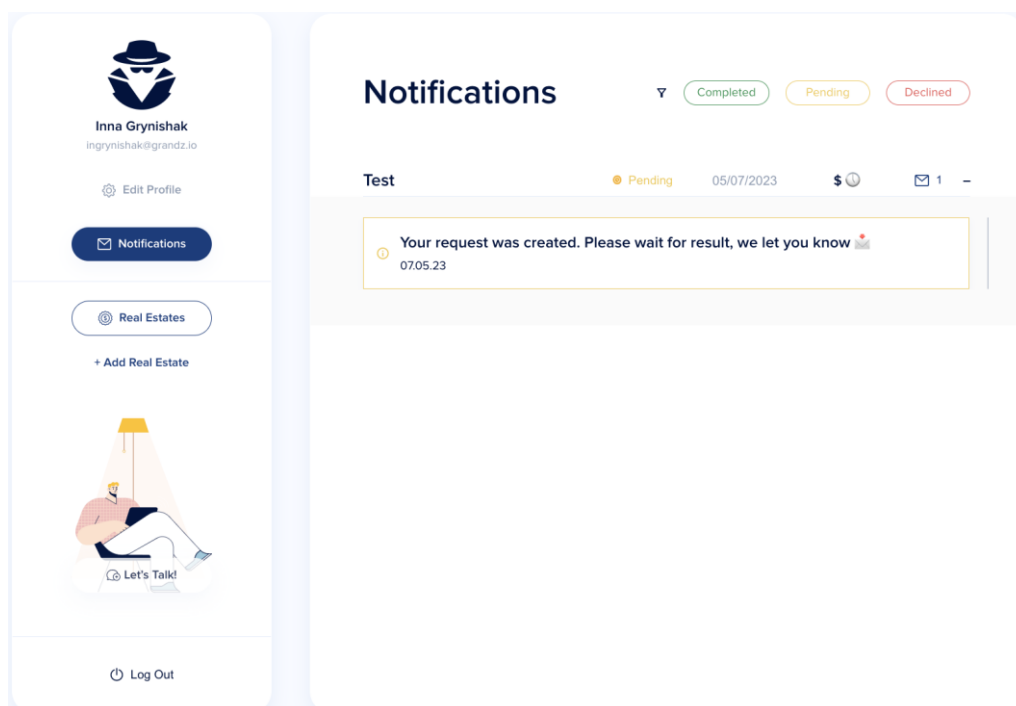


Рисунок 3.10 — Сторінка повідомлень про зміни оцінки об'єктів нерухомості

Як тільки результат буде готовий, користувач отримає відповідний лист (рисунок 3.11), а також з'явиться відповідне повідомлення на сторінці “Notifications” (рисунок 3.12), та оновиться інформація на сторінці “Requests”(рисунок 3.13) (представлено результати, у випадку коректних даних про ОН від користувача, в іншому випадку буде статус «Declined»).

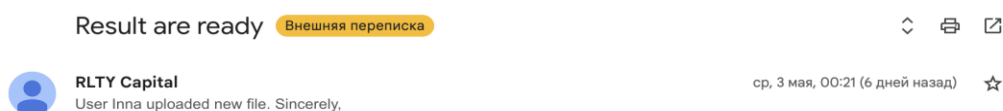


Рисунок 3.11 — Лист про закінчення процесу оцінки

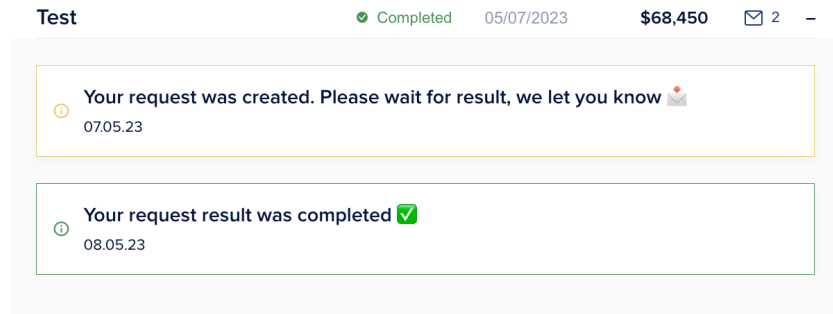


Рисунок 3.12 — Сторінка повідомлень про зміни об'єктів нерухомості

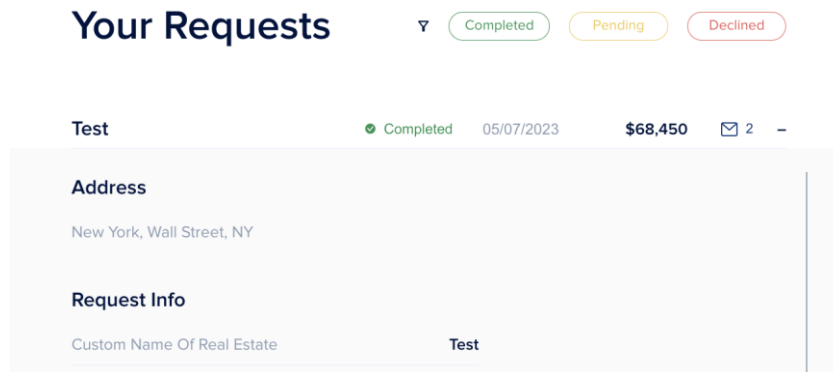


Рисунок 3.13 — Зміни на сторінці всіх об'єктів нерухомості

Окрім основної функціональності сайту, у боковому меню є ще кнопка “Edit Profile”, яка відкриває сторінку з налаштуваннями та інформацією про користувача (рисунок 3.14).

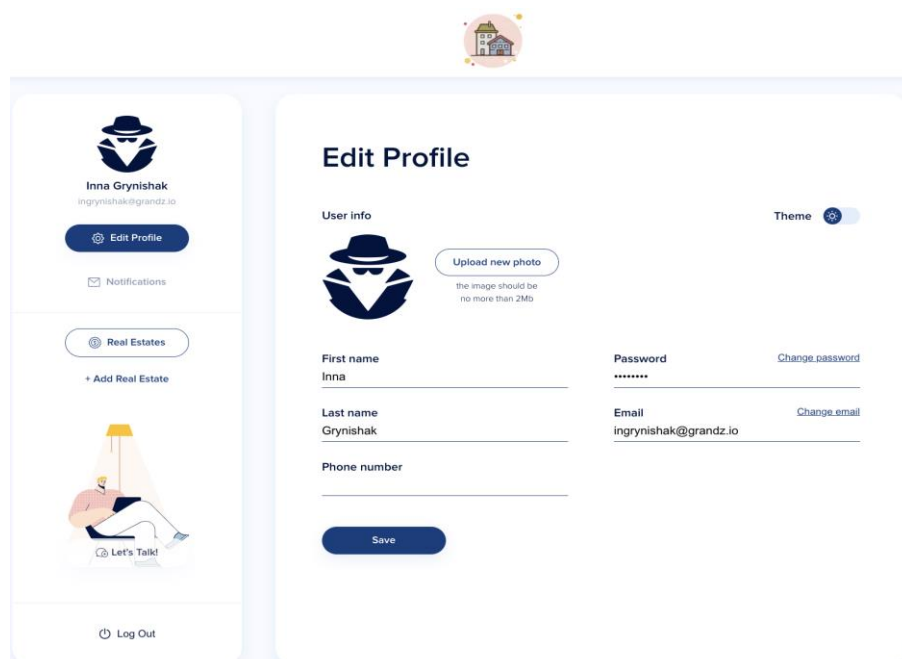


Рисунок 3.14 — Сторінка «Edit Profile»

На рисунку 3.14 є можливість змінити тему “Theme” з денної на нічну, та навпаки. Результат зміни на нічну тему представлено на рисунку 3.15.

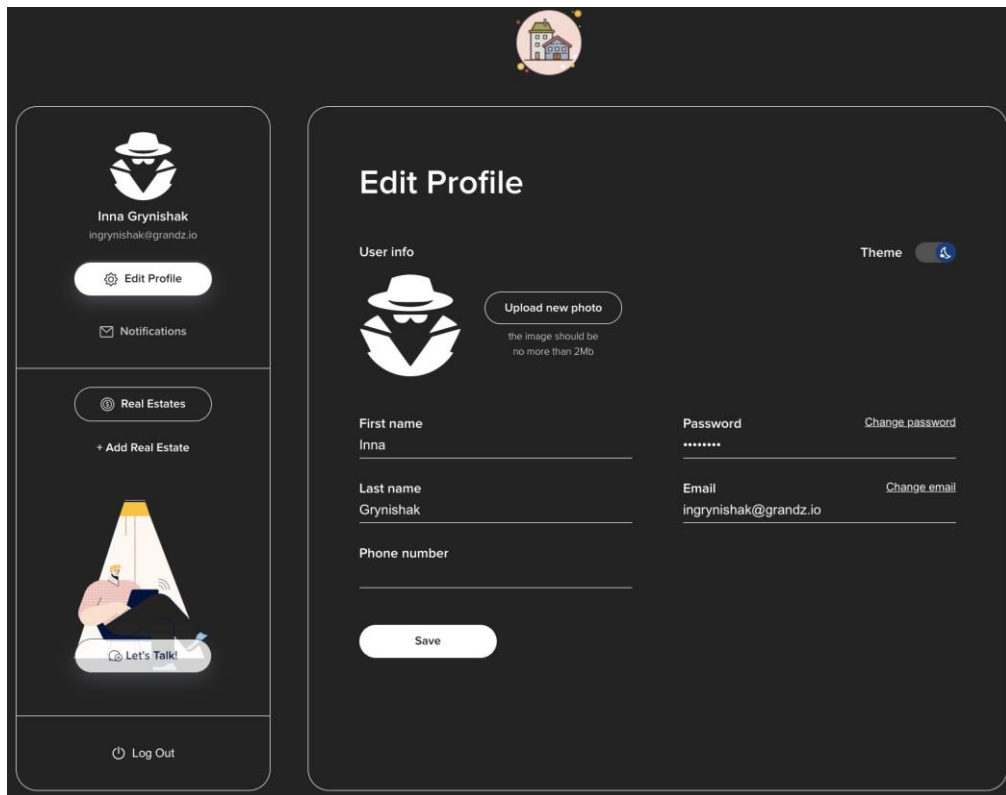


Рисунок 3.15 — Приклад сторінки з нічною темою

Також на даній сторінці можна завантажити фото профілю(рисунок 3.16).

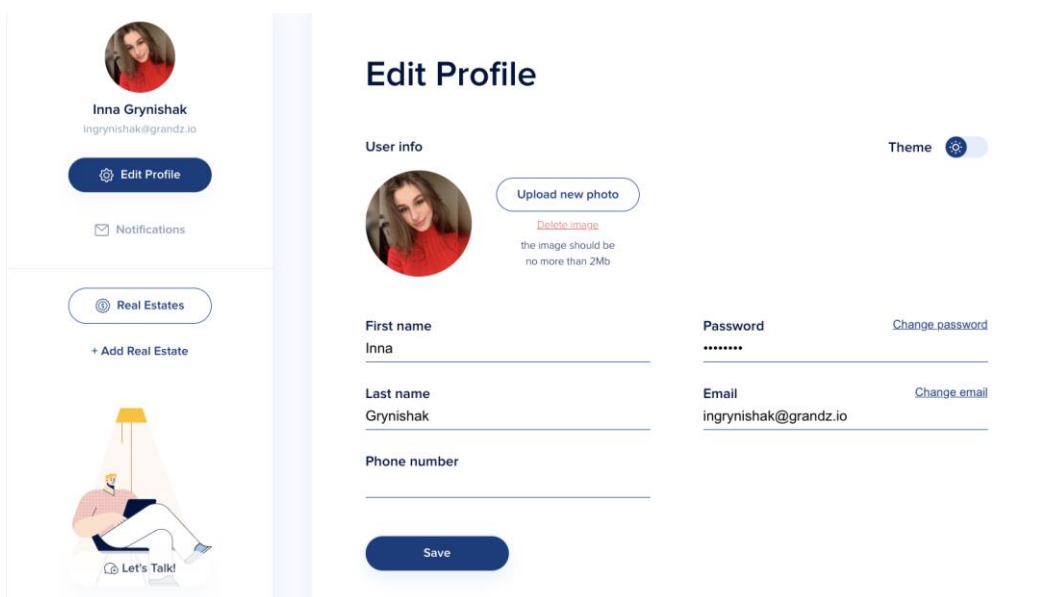


Рисунок 3.16 — Приклад профілю користувача із фото

Більше того є можливість змінити пароль, для цього потрібно натиснути “Change password”, як наслідок з’являється модальне вікно з проханням ввести поточний пароль та двічі новий для підтвердження(присутня перевірка на валідність даних)(рисунок 3.17).

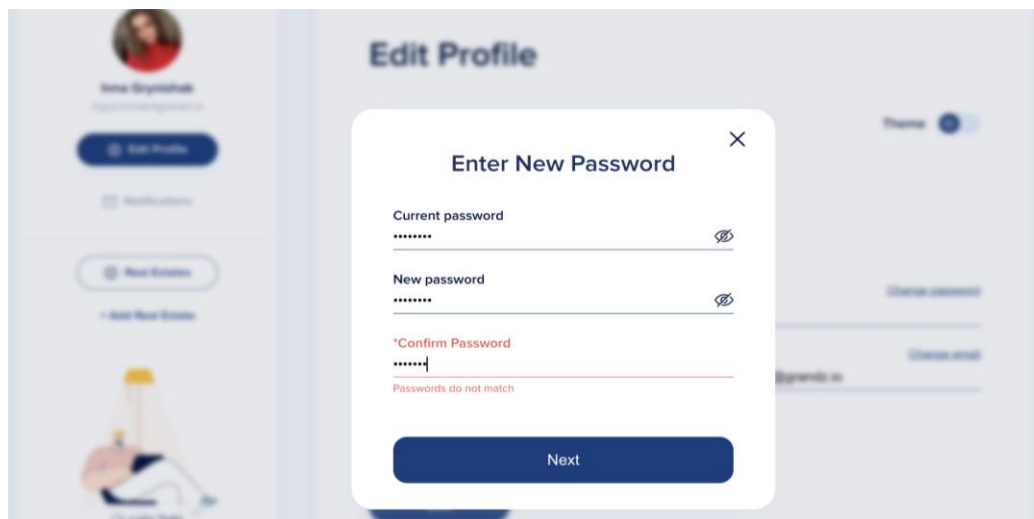
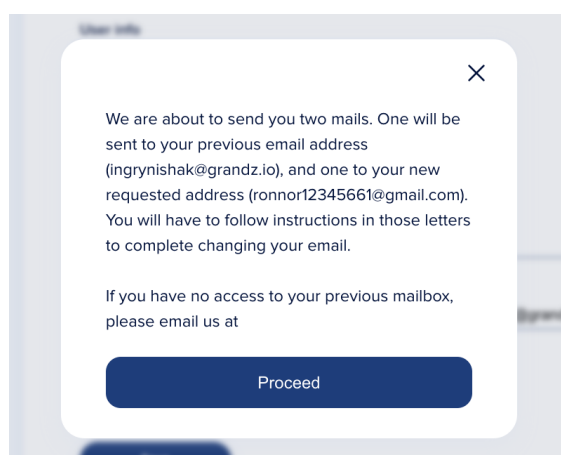
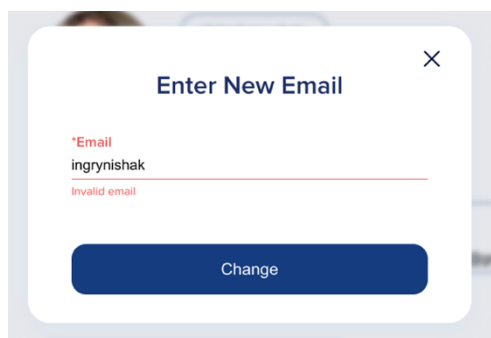


Рисунок 3.17 — Модальне вікно для зміни поточного паролю

Також є можливість змінити адресу електронної пошти, для цього потрібно натиснути “Change email”, як наслідок з’являється модальне вікно з проханням ввести email, на який планується поміняти поточний email(рисунки 3.18 – 3.19).



Рисунки 3.18 – 3.19 — Початкові етапи процесу зміни пошти

Після цього на пошту приходять лист з посиланням на сторінку системи з можливістю підтвердження зміни пошти.

3.3.2 Керівництво користувача-експерта

Другий тип користувачів системи — це користувач-експерт, чия робота полягає в аналізі даних про об'єкт, які надав користувач, у внесенні додаткової інформації та в проставленні пріоритетів. Після надання цих даних системі, запускається процес оцінки відповідного об'єкта нерухомості та як результат цього процесу формуються звіти про оцінювання.

Розглянемо весь процес, який експерт проходить під час своєї роботи. Отже, процес починається з того, що експерта інформують електронним листом про новий ОН. Лист містить посилання, яке перенаправляє експерта на ReTool, де його зустрічає вікно, зображене на рисунку 3.20.

Real estates for calculation

custom name	area	count of room	floor	balcony	condition	build condition	street	city	state	date of creation
Test	55	2	3	✓	Good 😊	Good 😊	Wall Street	New York	New York	May 24, 2023

Showing 1-1 of 1

👋 Hello Inna!

Please add additional information about the location of the real estate object (where 10 is the best and 1 is the worst).

To metro:


To center:

To supermarket:

City:

State:

Error in user data



© Mapbox © OpenStreetMap Improve this map

Please set priorities for now. The scale has 9 values, where 1 means that the criterion has the same importance as other criteria, 3 - has weak importance, 5 - noticeably weak, 7 - very weak, 9 - extremely weak, 2 - has significantly greater importance, 4 - has higher value, 6 - very important, 8 - extremely important.

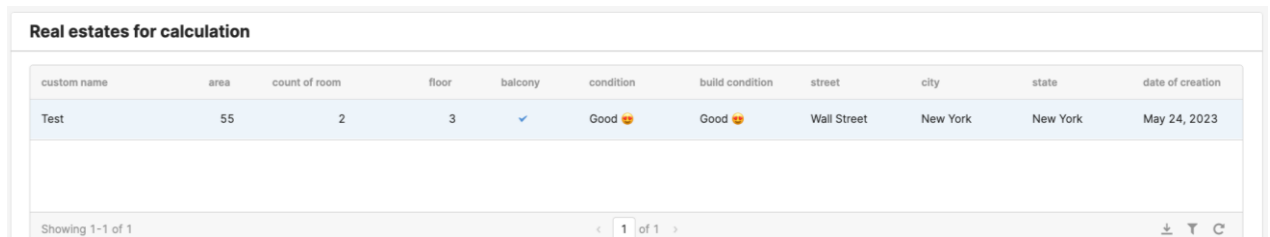
Set priorities

	Balcony	Floor	Count of room	To supermarket	To center	To metro	State	City	Area	Build condition	Condition
Balcony	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
Floor	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
Count of room	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
To supermarket	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
To center	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
To metro	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
State	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
City	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
Area	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
Build condition	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
Condition	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>

Finish

Рисунок 3.20 — Сторінка для роботи експерта

Давайте детальніше розглянемо це вікно. Експерт починає свою роботу з перегляду списку всіх ОН в системі, які потребують його розгляду експерта (рисунок 3.21). Експерт повинен обрати рядок таблиці, що відповідає ОН, над яким він збирається працювати. За замовчуванням, обрано перший рядок таблиці, проте користувач може обрати будь-який інший з списку.



custom name	area	count of room	floor	balcony	condition	build condition	street	city	state	date of creation
Test	55	2	3	✓	Good 🟡	Good 🟡	Wall Street	New York	New York	May 24, 2023

Showing 1-1 of 1

Рисунок 3.21 — Список об'єктів нерухомості доступні для розгляду експерта

Відповідно до обраного ОН на попередньому етапі, наступні компоненти та дані стосуватимуться цього об'єкту. Так, наступним компонентом є легко масштабована мапа з міткою місця розташування обраного ОН, яка надається експерту для глибшого аналізу місцезнаходження. Якщо експертом не було виявлено ніяких проблем у наданій інформації про ОН, то він повинен внести дані про необхідні показники (рисунок 3.22). В іншому випадку, експерт повинен вказати на наявність помилки в даних за допомогою кнопки «Error in user data», після чого з'явиться повідомлення про успішне завершення роботи над даним ОН з проханням перезавантажити сторінку (рисунок 3.23). Паралельно цьому, система проставить запиту статус «Declined», тому при перезавантаженні сторінки дані про цей об'єкт не будуть відображені в таблиці з рисунку 3.21.



Hello Innal

Please add additional information about the location of the real estate object (where 10 is the best and 1 is the worst).

To metro: 1

To center: 1

To supermarket: 1

City: 1

State: 1

Error in user data

mapbox

© Mapbox © OpenStreetMap Improve this map

Рисунок 3.22 — Компонент сторінки, який відповідає за збір даних та мапу

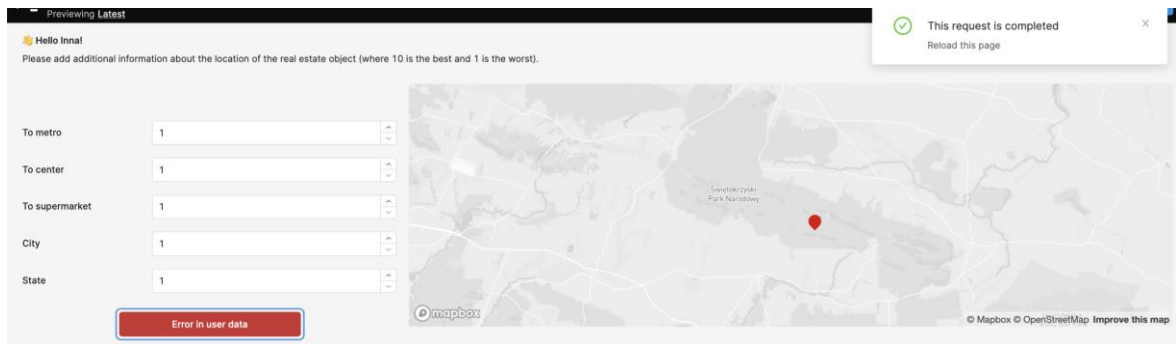


Рисунок 3.23 — Результат натискання кнопки «Error in user data»

Якщо дані про об'єкт нерухомості були коректні, то користувач-експерт переходить до фінального етапу своєї роботи — проставлення пріоритетів. Для цього йому надається відповідна таблиця попарних порівнянь пріоритетів та нагадуються правила її заповнення (рисунок 3.24). В таблиці, експерт заповнює лише значення верхнього правого трикутника, а решту значень генерує система автоматично, що забезпечує економію часу та дотримання правила про симетричні пари.

Please set priorities for now. The scale has 9 values, where 1 means that the criterion has the same importance as other criteria, 3 - has weak importance, 5 - noticeably weak, 7 - very weak, 9 - extremely weak, 2 - has significantly greater importance, 4 - has higher value, 6 - very important, 8 - extremely important.

Set priorities

	Balcony	Floor	Count of room	To supermarket	To center	To metro	State	City	Area	Build condition	Condition
Balcony	1	1	1	1	1	1	1	1	1	1	1
Floor	1	1	1	1	1	1	1	1	1	1	1
Count of room	1	1	1	1	1	1	1	1	1	1	1
To supermarket	1	1	1	1	1	1	1	1	1	1	1
To center	1	1	1	1	1	1	1	1	1	1	1
To metro	1	1	1	1	1	1	1	1	1	1	1
State	1	1	1	1	1	1	1	1	1	1	1
City	1	1	1	1	1	1	1	1	1	1	1
Area	1	1	1	1	1	1	1	1	1	1	1
Build condition	1	1	1	1	1	1	1	1	1	1	1
Condition	1	1	1	1	1	1	1	1	1	1	1

Finish

Рисунок 3.24 — Таблиця для проставлення пріоритетів

При натисканні на кнопку «Finish» відразу запускається процес перевірки введених пріоритетів на узгодженість. У випадку провалу цієї перевірки, експерт отримує повідомлення про помилку з проханням проаналізувати проставлені ним пріоритети та внести в них корективи (рисунок 3.25).

Consistency check is failed
Please, analyze the previous values and prioritize again, thanks!

significantly greater importance, 4 - has higher value, 6 - very important, 8 - extremely important.

Set priorities

	Balcony	Floor	Count of room	To supermarket	To center	To metro	State	City	Area	Build condition	Condition
Balcony	1	3	9	1	1	1	1	1	1	1	1
Floor	0,333	1	1	9	1	1	1	1	1	1	1
Count of room	0,111	1	1	2	1	1	1	1	1	1	1
To supermarket	1	0,111	0,5	1	1	8	1	1	1	1	1
To center	1	1	1	1	1	1	1	1	1	1	1
To metro	1	1	1	0,125	1	1	1	1	1	1	1

Рисунок 3.25 — Повідомлення при провальній перевірці на консистентність

В разі успішної перевірки, користувач-експерт отримує повідомлення про завершення роботи над розглядуваним об'єктом нерухомості (рисунок 3.26).

Everything is finished
Please, reload this page, thanks!

Error in user data

Please set priorities for now. The scale has 9 values, where 1 means that the criterion has the same importance as other criteria, 3 - has weak importance, 5 - noticeably weak, 7 - very weak, 9 - extremely weak, 2 - has significantly greater importance, 4 - has higher value, 6 - very important, 8 - extremely important.

Set priorities

	Balcony	Floor	Count of room	To supermarket	To center	To metro	State	City	Area	Build condition	Condition
Balcony	1	2	7	3	8	4	9	8	8	9	9
Floor	0,5	1	5	4	7	5	9	5	7	9	9
Count of room	0,143	0,2	1	1	2	2	4	4	7	8	7
To supermarket	0,333	0,25	1	1	2	2	5	4	6	7	7
To center	0,125	0,143	0,5	0,5	1	1	3	4	5	5	5
To metro	0,25	0,2	0,5	0,5	1	1	3	4	4	3	3

Рисунок 3.26 — Повідомлення про завершення роботи експерта

Таким чином, на цьому моменті робота експерта з об'єктом нерухомості завершена. Якщо з'являться нові об'єкти, експерт буде проінформований електронною поштою і процес, який було описано вище, буде повторено.

3.4 Тестування системи визначення оціночної вартості нерухомості

Процес тестування є важливою складовою розробки програмного забезпечення, адже він допомагає забезпечити високу якість продукту, перевірити розроблену систему на відповідність вимогам та очікуванням користувачів, зменшити ризики виникнення помилок, виявити та виправити усі можливі недоліки програми тощо.

Розглядаючи розроблену систему визначення оціночної вартості нерухомості, першочергово варто розглянути процес тестування найважливішої задачі системи, яка стосується розрахунку оцінки. Дана задача розпочинається процесом збору даних від клієнта та їх базової перевірки на введення інтуїтивно правильних даних. Опис цього процесу міститься в розділі 3.3.1 "Керівництво користувача-клієнта". В цьому розділі були наведені приклади відповідей програми у випадку некоректно заповненої форми (наприклад, коли кількість кімнат від'ємна або обов'язкове поле порожнє). Таким чином, система не дозволить відправити форму з інтуїтивно некоректними даними. Глибший аналіз даних ОН буде проведено експертом, в результаті можливі три наступні сценарії:

1. Після успішної відправки форми з даними про ОН, в ході аналізу експертом були виявлені помилки чи деяка невідповідність даних.
2. Після успішної відправки форми з даними про ОН, в ході роботи експерта були проставлені не консистентні пріоритети.
3. Успішний процес, внаслідок якого, був наданий готовий результат.

Розглянемо перший сценарій. Нехай ОН описується даними з таблиці 3.3:

Таблиця 3.3 — Опис даних ОН для тестування першого сценарію

Назва критерію	Значення критерію
Кастомна назва ОН	Real Estate with bad input data
Загальна площа в квадратних метрах	55 sq.m.
Кількість кімнат	1
Вулиця	Oakland Street
Місто	Manchester
Штат	Connecticut
Широта, довгота	23.659471, -53.973770
Поверх	4
Наявність балкону	+
Стан квартири	Good 😊
Стан відповідного будинку	Average 😊
Бажана дата покупки / продажу	19.05.2023

На перший погляд дані виглядають коректними, проте при глибшому аналізі місця розташування можна побачити, що задані широта і довгота вказують на місце, яке не відповідає вказаній адресі і де взагалі неможливе знаходження квартири (рисунок 3.27). Тому, як результат, ми повинні закінчити процес роботи над цим ОН, проставити статус запиту «Відмінений» і проінформувати про це користувача.

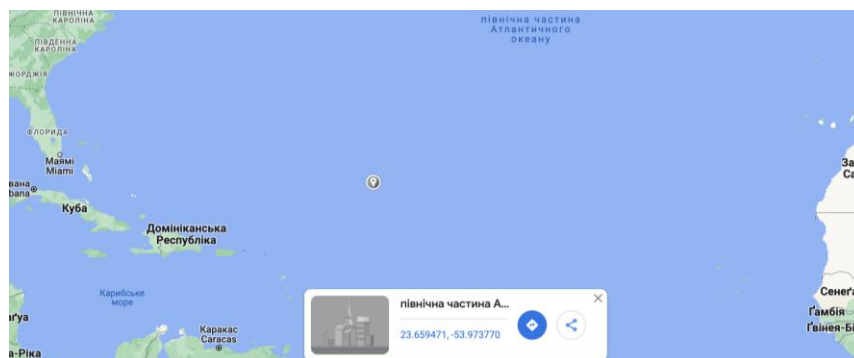


Рисунок 3.27 — Відображення на карті вказаних значень широти і довготи

Перевіримо результат, який видасть програма, при описаних вище ввідних даних, що зумовлюють попадання в перший сценарій (рисунок 3.28).

<p>Custom name of Real Estate Real Estate with bad input data</p> <hr/> <p>Area in sq.m. 55</p> <hr/> <p>Count of room 1</p> <hr/> <p>Street Address Oakland Street</p> <hr/> <p>City Manchester</p> <hr/> <p>State Connecticut</p> <hr/> <p>Longitude -53.973770</p> <hr/> <p>Latitude 23.659471</p> <hr/> <p>Next</p>	<p style="text-align: right;">< Back</p> <p>Floor 4</p> <hr/> <p><input checked="" type="checkbox"/> Balcony</p> <p>Condition of apartment Good 😊</p> <hr/> <p>Condition of building Average 😊</p> <hr/> <p>Date You Expect to sell or to buy 19.05.2023</p> <hr/> <p>Add</p>
---	--

Рисунок 3.28 — Ввід даних, що зумовлюють попадання в перший сценарій

У результаті відправки форми, приходять повідомлення про успішно створений запит на розрахунок оціночної вартості введеного ОН (рисунок 3.29), і після цього розпочинається робота експерта в ReTool (рисунок 3.30).

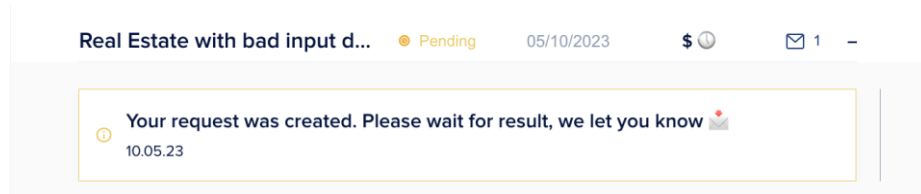


Рисунок 3.29 — Повідомлення про успішно створений запит

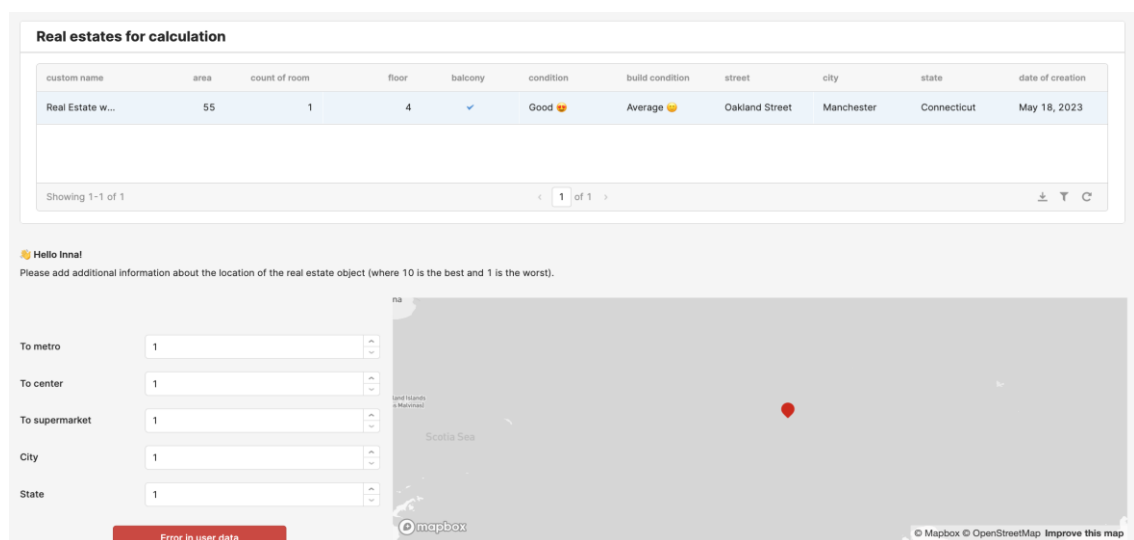


Рисунок 3.30 — Вигляд даних ОН, що тестується, в ReTool

Мапа з рисунку 3.30 відображує некоректність даних, внаслідок чого експерт натискає кнопку «Error in user data». В результаті, система проставляє запити статус «Declined»(тобто відмінений) і інформує користувача про зміну статусу запити повідомленням у користувацькому інтерфейсі (рисунок 3.31).

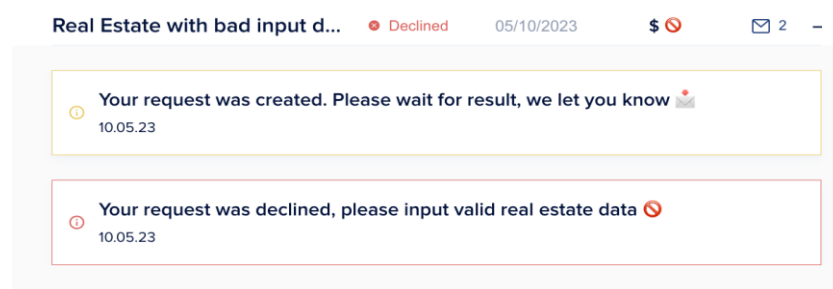


Рисунок 3.31 — Повідомлення про зміну статусу запити на «Declined»

Аналізуючи отримані програмою дані, варто сказати, що програма видала очікуваний та бажаний результат. В наступній версії продукту, пропонується реалізація мапи для збору даних від користувача, яка зможе усунути можливість виникнення різних проблем з заданням місця розташування об'єкту.

Розглянемо два інші сценарії, нехай об'єкт нерухомості описується даними з таблиці 3.4:

Таблиця 3.4 — Опис даних ОН для тестування двох інших сценаріїв

Назва критерію	Значення критерію
Кастомна назва ОН	Real Estate with good input data
Загальна площа в квадратних метрах	55 sq.m.
Кількість кімнат	1
Вулиця	Morgan Street
Місто	Hartford
Штат	Connecticut
Широта, довгота	41.77043, -72.67195
Поверх	3
Наявність балкону	+
Стан квартири	Good 😍
Стан відповідного будинку	Average 😊
Бажана дата покупки / продажу	19.05.2023

В цій ситуації, при аналізі місця розташування можна побачити, що задані широта і довгота вказують на коректне місце (рисунок 3.32).

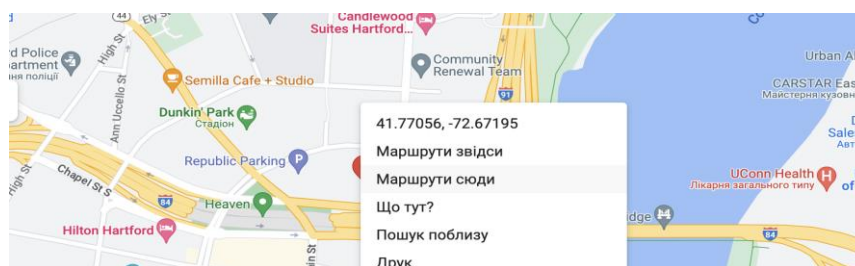


Рисунок 3.32 — Відображення на карті вказаних значень широти і довготи

Завдяки мапі зображеній на рисунку 3.32, проставимо значення необхідних нам критеріїв від 1 до 10, де 10 це найкраща оцінка, 1 – найгірша: близькість до метро – 10, до центру – 9, до супермаркету – 8, загальна оцінка штату – 8, загальна оцінка міста – 9.

На цьому моменті, можемо переходити до етапу заповнення матриці попарних порівнянь пріоритетів. Першим розглянемо випадок, коли матриця не консистентна, нехай маємо матрицю задану таблицею 3.5 (де Б – балкон, П – поверх, К – кількість кімнат, С – до супермаркету, Ц – до центру, М – до метро, Ш – штат, Мс – місто, Пл – площа, Сб – стан будинку, Ск – стан квартири)

Таблиця 3.5 — Матриця попарних порівнянь пріоритетів

	Б	П	К	С	Ц	М	Ш	Мс	Пл	Сб	Ск
Б	1	3	9	1	1	1	1	1	1	1	1
П	1 / 3	1	1	9	1	1	1	1	1	1	1
К	1 / 9	1	1	2	1	1	1	1	1	1	1
С	1	1 / 9	0 / 5	1	1	8	1	1	1	1	1
Ц	1	1	1	1	1	1	1	1	1	1	1
М	1	1	1	1 / 8	1	1	1	1	1	1	1
Ш	1	1	1	1	1	1	1	1	2	1	1
Мс	1	1	1	1	1	1	1	1	1	1	1
Пл	1	1	1	1	1	1	1 / 2	1	1	1	1
Сб	1	1	1	1	1	1	1	1	1	1	1
Ск	1	1	1	1	1	1	1	1	1	1	1

Далі, за алгоритмом методу аналізу ієрархій, відбувається розрахунок коефіцієнтів вагомості критеріїв. Для цього розраховується власний вектор матриці представленої таблицею 3.5. Кожен елемент вектору W обчислюється як середнє геометричне відповідного рядка матриці:

$$w_1 = \sqrt[11]{1 * 3 * 9 * 1^8} = 1.349348$$

$$w_2 = \sqrt[11]{0.333 * 9} = 1.104931$$

$$w_3 = \sqrt[11]{2 * 0.111} = 0.872122$$

$$w_4 = \sqrt[11]{0.2 * 0.125 * 8} = 0.863887$$

$$w_5 = w_8 = w_{10} = w_{11} = \sqrt[11]{1} = 1$$

$$w_6 = \sqrt[11]{0.125} = 0.827753$$

$$w_7 = \sqrt[11]{2} = 1.065041$$

$$w_9 = \sqrt[11]{0.5} = 0.9389309$$

Обрахуємо середнє арифметичне:

$$\frac{1.349348 + 1.104931 + 0.872122 + 0.863887 + 0.827753 + 1.065041 + 0.9389309 + 4}{11} = 1.002001173$$

Отже, елементи власного вектору матриці, заданою таблицею 3.5, наступні:

$$w_1 = \frac{1.349348}{1.002001173} = 1.746653114$$

$$w_2 = \frac{1.104931}{1.002001173} = 1.602724258$$

$$w_3 = \frac{1.00872122}{1.002001173} = 1.08703802$$

$$w_4 = \frac{0.863887}{1.002001173} = 0.862161665$$

$$w_5 = \frac{1}{1.002001173} = 0.459471694$$

$$w_6 = \frac{0.827753}{1.002001173} = 0.826099831$$

$$w_7 = \frac{1.065041}{1.002001173} = 1.062913925$$

$$w_8 = \frac{1}{1.002001173} = 0.459471694$$

$$w_9 = \frac{0.9389309}{1.764126369} = 0.651412171$$

$$w_{10} = \frac{1}{2.176412636} = 0.459471694$$

$$w_{11} = \frac{1}{1.764126369} = 0.659471694$$

Після розрахунку ваг важливості, наступним етапом МАІ є перевірка на консистентність. Виконаємо її за алгоритмом описаним у розділі 2.1.5.

Спочатку обрахуємо максимальне власне значення λ_{max} :

$$\begin{aligned}\lambda_{max} = & 1.746653114 + 1.602724258 + 1.08703802 + 0.862161665 + 0.459471694 \\ & + 0.459471694 + 0.459471694 + 0.459471694 + 0.459471694 \\ & + 1.824521203 + 2.590402623 = 12.854486\end{aligned}$$

Наступним кроком розрахуємо індекс консистентності T :

$$T = \frac{|\lambda_{max} - n|}{n - 1} = \frac{12.854486 - 11}{10} = 0.18544864$$

І нарешті обчислимо відношення узгодженості як відношення обчисленого індексу узгодженості T до табличного індексу $M(T)$ з таблицьки 2.2, яка знаходиться в розділі 2.1.5.

$$U = \frac{T}{M(T)} = \frac{0.18544864}{1.52} = 0.122 \quad (3.1)$$

Внаслідок перевірки на консистентність, розглянута матриця не є узгодженою ($\frac{T}{M(T)} < 0.1$), враховуючи це, система повинна проінформувати про це експерта, і вказати на потребу в повторному заповненні таблиці попарних порівнянь пріоритетів.

Перевіримо результат, який видасть розроблена система визначення оціночної вартості нерухомості, при описаних вище ввідних даних (рисунок 3.23). У результаті відправки форми з описаними даними, знову ж приходить повідомлення про успішно створений запит на розрахунок оцінки введеного ОН (рисунок 3.24).

Рисунок 3.23 — Ввід даних, що зумовлюють попадання в два інших сценарії

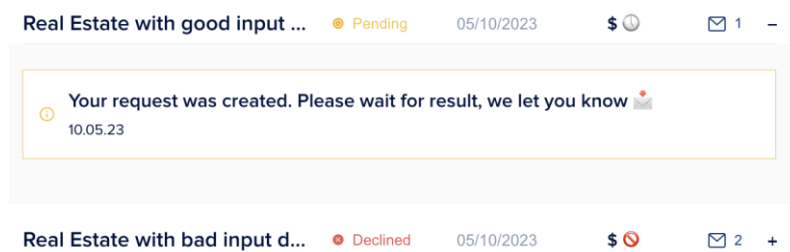


Рисунок 3.24 — Повідомлення про успішно створений запит

Після повідомлення про успішно створений запит, розпочинається робота експерта в ReTool(рисунок 3.25). Мапа з даного рисунку підтверджує коректність даних, і завдяки неї проставляємо необхідні дані аналогічні тим, що були описані вище.

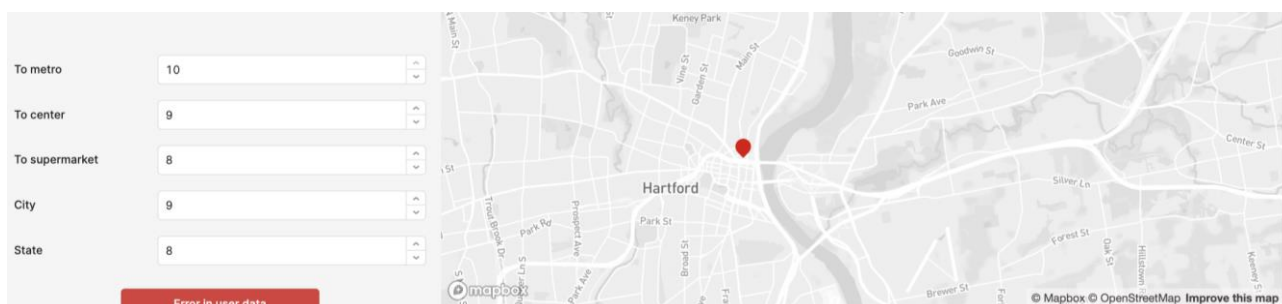


Рисунок 3.25 — Ввід необхідних даних про ОН експертом

Враховуючи те, що при глибошому аналізі не було виявлено ніяких проблем у наданій інформації про ОН, переходимо до заповнення таблиці попарних порівнянь пріоритетів, вносимо дані з таблиці 3.5(рисунок 3.26).

Set priorities											
	Balcony	Floor	Count of room	To supermarket	To center	To metro	State	City	Area	Build condition	Condition
Balcony	1	3	9	1	1	1	1	1	1	1	1
Floor	0,333	1	1	9	1	1	1	1	1	1	1
Count of room	0,111	1	1	2	1	1	1	1	1	1	1
To supermarket	1	0,111	0,5	1	1	8	1	1	1	1	1
To center	1	1	1	1	1	1	1	1	1	1	1
To metro	1	1	1	0,125	1	1	1	1	1	1	1
State	1	1	1	1	1	1	1	1	2	1	1
City	1	1	1	1	1	1	1	1	1	1	1
Area	1	1	1	1	1	1	0,5	1	1	1	1
Build condition	1	1	1	1	1	1	1	1	1	1	1
Condition	1	1	1	1	1	1	1	1	1	1	1

Рисунок 3.26 — Вигляд таблиці попарних порівнянь пріоритетів

На основі даних з рисунку 3.26 відбувається розрахунок ваг та їх перевірка на консистентність. Відповідно до розрахунків виконаних вище (3.1), ваги не є консистентні, тому система повинна видати відповідне повідомлення експерту (рисунок 3.27).

Previewing Latest
significantly greater importance, 4 - has higher value, 6 - very important, 8 - extremely important.

Consistency check is failed

Please, analyze the previous values and prioritize again, thanks!

Set priorities											
	Balcony	Floor	Count of room	To supermarket	To center	To metro	State	City	Area	Build condition	Condition
Balcony	1	3	9	1	1	1	1	1	1	1	1
Floor	0,333	1	1	9	1	1	1	1	1	1	1
Count of room	0,111	1	1	2	1	1	1	1	1	1	1
To supermarket	1	0,111	0,5	1	1	8	1	1	1	1	1
To center	1	1	1	1	1	1	1	1	1	1	1
To metro	1	1	1	0,125	1	1	1	1	1	1	1

Рисунок 3.27 — Очікуваний результат

Таким чином, відповідно до інструкцій повідомлення, наведеного на рисунку 3.27, внесемо правки до таблиці попарних порівнянь пріоритетів (таблиця 3.6), щоб перейти до останнього, третього сценарію.

Таблиця 3.6 — виправлена таблиця попарних порівнянь пріоритетів

	Б	П	К	С	Ц	М	Ш	Мс	Пл	Сб	Ск
Б	1	2	3	5	5	7	7	7	8	8	9
П	0.5	1	3	5	5	7	7	7	8	8	9
К	0.333	0.333	1	2	3	5	7	7	7	8	9
С	0.2	0.2	0.5	1	2	4	6	6	7	7	8
Ц	0.2	0.2	0.333	0.5	1	3	5	6	6	6	8
М	0.143	0.143	0.2	0.25	0.333	1	4	5	6	6	8
Ш	0.143	0.143	0.143	0.167	0.2	0.25	1	2	3	5	7
Мс	0.143	0.143	0.143	0.167	0.167	0.2	0.5	1	3	4	7
Пл	0.125	0.125	0.143	0.143	0.167	0.167	0.333	0.333	1	3	5
Сб	0.125	0.125	0.125	0.143	0.167	0.167	0.2	0.25	0.33	1	2
Ск	0.111	0.111	0.111	0.125	0.125	0.125	0.143	0.143	0.2	0.5	1

Знову пройдемо описані вище процеси обрахунку ваг важливості та їх перевірки на консистентність.

Розраховуємо ваги важливості, для цього розрахуємо власний вектор W матриці, яка представлена таблицею 3.6.

$$w_1 = \sqrt[11]{1 * 0.5 * 0.33 * 0.2 * 0.2 * 0.143 * 0.143 * 0.143 * 0.125 * 0.125 * 0.11} = 0.700726$$

$$w_2 = \sqrt[11]{2 * 1 * 0.333 * 0.2 * 0.2 * 0.143 * 0.143 * 0.143 * 0.125 * 0.125 * 0.111} = 0.794842$$

$$w_3 = \sqrt[11]{3 * 3 * 1 * 0.5 * 0.333 * 0.2 * 0.143 * 0.143 * 0.143 * 0.125 * 0.111} = 0.990467$$

$$w_4 = \sqrt[11]{5 * 5 * 2 * 1 * 0.5 * 0.25 * 0.167 * 0.167 * 0.143 * 0.143 * 0.125} = 1.181280$$

$$w_5 = \sqrt[11]{5 * 5 * 3 * 2 * 1 * 0.333 * 0.2 * 0.167 * 0.167 * 0.167 * 0.125} = 1.362334$$

$$w_6 = \sqrt[11]{7 * 7 * 5 * 4 * 3 * 1 * 0.25 * 0.2 * 0.167 * 0.167 * 0.125} = 1.574087$$

$$w_7 = \sqrt[11]{7 * 7 * 7 * 6 * 5 * 4 * 1 * 0.5 * 0.333 * 0.2 * 0.143} = 2.131026$$

$$w_8 = \sqrt[11]{7 * 7 * 7 * 6 * 6 * 5 * 2 * 1 * 0.333 * 0.25 * 0.143} = 2.559409$$

$$w_9 = \sqrt[11]{8 * 8 * 7 * 7 * 6 * 6 * 3 * 3 * 1 * 0.333 * 0.2} = 3.037673$$

$$w_{10} = \sqrt[11]{8 * 8 * 8 * 7 * 6 * 6 * 5 * 4 * 3 * 1 * 0.5} = 3.970911$$

$$w_{11} = \sqrt[11]{9 * 9 * 9 * 8 * 8 * 8 * 7 * 7 * 5 * 2 * 1} = 5.637785$$

Обрахуємо середнє арифметичне для наступного процесу нормування:

$$\frac{0.700726 + 0.794842 + 0.990467 + 1.181280 + 1.362334 + 1.574087 + 2.131026 + 2.559409 + 3.037673 + 3.970911}{11} = 2.176412636$$

Отже, власний вектор W матриці, заданою таблицею 3.5, складається з:

$$w_1 = \frac{0.700726}{2.176412636} = 0.3219637621$$

$$w_2 = \frac{0.794842}{2.176412636} = 0.3652074$$

$$w_3 = \frac{0.990467}{2.176412636} = 0.45509155$$

$$w_4 = \frac{1.181280}{2.176412636} = 0.542764722$$

$$w_5 = \frac{1.362334}{2.176412636} = 0.62595391$$

$$w_6 = \frac{1.5740871}{2.176412636} = 0.723248466$$

$$w_7 = \frac{2.131026}{2.176412636} = 0.979146125$$

$$w_8 = \frac{2.559409}{2.176412636} = 1.175975988$$

$$w_9 = \frac{3.037673}{2.176412636} = 1.395724758$$

$$w_{10} = \frac{3.970911}{2.176412636} = 1.824521203$$

$$w_{11} = \frac{5.637785}{2.176412636} = 2.590402623$$

Наступним кроком виконаємо перевірку на консистентність. Спочатку обрахуємо максимальне власне значення λ_{max} :

$$\begin{aligned} \lambda_{max} &= 0.3219637621 + 0.3652074 + 0.45509155 + 0.542764722 + 0.62595391 \\ &\quad + 0.723248466 + 0.979146125 + 1.175975988 + 1.395724758 \\ &\quad + 1.824521203 + 2.590402623 = 11.0000005071 \end{aligned}$$

Наступним кроком розрахуємо індекс консистентності T :

$$T = \frac{|\lambda_{max} - n|}{n - 1} = \frac{11.0000005071 - 11}{10} = 0.00000005071$$

І нарешті обчислимо відношення узгодженості:

$$U = \frac{T}{M(T)} = \frac{0.00000005071}{1.52} = 0.00000003336 \quad (3.2)$$

Внаслідок перевірки на консистентність, розглянута матриця є узгодженою ($0.00000003336 < 0.1$). Враховуючи це, можемо перейти до останнього етапу МАІ — розрахунку оціночної вартості ОН. Для цього використаємо зважене сумування, використовуючи попередньо розраховані ваги критеріїв та їх значення:

$$\begin{aligned} Result &= (0.3219637621 * 1 + 0.3652074 * 3 + 0.45509155 * 1 + \\ &0.542764722 * 8 + 0.62595391 * 10 + 0.723248466 * 9 + 0.979146125 * 9 + \\ &1.175975988 * 10 + 1.395724758 * 55 + 1.824521203 * 8 + 2.590402623 * 6) * \\ &2000 = 68\ 400 \end{aligned}$$

Таким чином, система повинна отримати оціночну вартість ОН рівною шістдесят вісім тисяч чотиреста доларів та сформувані звіти про отриману оцінку й передати їх користувачу - клієнту.

Перевіримо результат, який видасть програма, з тими ж даними, що використовувались в другому сценарії, тільки змінивши таблицю попарних порівнянь пріоритетів відповідно до таблиці 3.6. Відповідно до проведених розрахунків вище (3.2), система повинна видати повідомлення про успішне завершення роботи експерта над даним ОН (рисунок 3.28), та на основі усієї доступної інформації розрахувати оцінку та проінформувати користувача про результати поштою, а також відповідним змінами у користувацькому інтерфейсі (рисунок 3.29).

	Balcony	Floor	Count of room	To supermarket	To center	To metro	State	City	Area	Build condition	Condition
Balcony	1	2	3	5	5	7	7	7	8	8	9
Floor	0,5	1	3	5	5	7	7	7	8	8	9
Count of room	0,333	0,333	1	2	3	5	7	7	7	8	9
To supermarket	0,2	0,2	0,5	1	2	4	6	6	7	7	8
To center	0,2	0,2	0,333	0,5	1	3	5	6	6	6	8
To metro	0,143	0,143	0,2	0,25	0,333	1	4	5	6	6	8
State	0,143	0,143	0,143	0,167	0,2	0,25	1	2	3	5	7
City	0,143	0,143	0,143	0,167	0,167	0,2	0,5	1	3	4	7
Area	0,125	0,125	0,143	0,143	0,167	0,167	0,333	0,333	1	3	5
Build condition	0,125	0,125	0,125	0,143	0,167	0,167	0,2	0,25	0,333	1	2
Condition	0,111	0,111	0,111	0,125	0,125	0,125	0,143	0,143	0,2	0,5	1

Рисунок 3.28 — Повідомлення про успішне завершення роботи над ОН

Test ✔ Completed 05/07/2023 \$68,450 ✉ 2 -

ⓘ Your request was created. Please wait for result, we let you know ✉
07.05.23

✔ Your request result was completed ✔
08.05.23

Рисунок 3.29 — Результат роботи системи

На основі результатів тестування можна зробити висновок, що розроблена система визначення оціночної вартості нерухомості є досить ефективною та коректною при всіх можливих сценаріях. Для покращення роботи системи при попаданні в перший сценарій, було запропоновано додати мапу для збору даних від користувача. Щодо двох інших сценаріїв, то для покращення точності результатів варто позбутись суб'єктивності оцінювання експертом, що можна зробити або запровадженням колективного експерту, або повністю автоматичним проставленням необхідних даних з використанням зібраної заздалегідь бази даних та інформації з зовнішніх ресурсів. Ці заходи можуть значно покращити якість результатів та забезпечити більшу надійність. Усі ці фактори свідчать про перспективність розробленої системи для використання в галузі оцінки нерухомості та можуть допомогти забезпечити більш об'єктивні та точні результати оцінювання.

3.5 Висновок

Даний розділ розпочався детальним обґрунтуванням вибраних програмних засобів для реалізації застосунку. В ході цього аналізу було описано переваги та доцільність використання того чи іншого засобу. Окрім цього, в даному розділі також було описано структуру програмного забезпечення, що включало в себе опис трьох основних рівнів: додатку, середовища виконання та операційної системи. Детальніше було розглянуто структуру рівня додатку, що включало аналіз як серверної так і клієнтської сторони застосунку. Для кожної сторони було наведено структурну схему відповідної системи з детальним описом та специфікацією основних програмних модулів у вигляді таблиць. Також, у підрозділі про структуру клієнтської сторони було використано граф переходів для демонстрації процесу переходу між сторінками в системі.

Після опису структури системи, було надано два окремі описи керівництв користувачів, оскільки в системі передбачено два типи користувачів — клієнт та експерт. Керівництво користувача-клієнта детально описує процес отримання конкретної оцінки, а також інші додаткові можливості, що доступні на сайті. Керівництво експерта описує весь процес, який експерт повинен пройти під час своєї роботи. Обидва керівництва містять детальну інформацію про особливості системи та описують різні можливі ситуації з наданням прикладів використання системи.

На завершення даного розділу, було описано процес тестування розробленої системи. Було проведено тестування різних сценаріїв виконання головної задачі системи — розрахунку оціночної вартості об'єкта нерухомості, проаналізовано результати та запропоновано шляхи їх покращень.

ВИСНОВОК

В ході виконання випускної кваліфікаційної роботи було спроектовано, реалізовано та перевірено на коректність автоматизовану систему визначення оціночної вартості нерухомості, з допомогою якої потенційні інвестори зможуть заощадити час та гроші для вибору вдалого інвестиційного об'єкту нерухомості.

Успішні результати тестування пояснюються збалансованим підходом виконання роботи, що включає три етапи: аналітичний етап, етап проектування та етап реалізації. В процесі аналітичного етапу було проведено глибокий аналіз особливостей задачі оцінки нерухомості (нормативно-правова база США, фактори, що впливають на оцінку, підходи до визначення оціночної вартості ОН та узагальнена технологія оцінки нерухомості), Окрім цього, було досліджено готові рішення вирішення задачі оцінки нерухомості, а також проведено порівняльний аналіз систем-аналогів розробленої системи. Щодо етапів проектування та реалізації, то було враховано всі особливості поставленої задачі та використано найкращі практики побудови веб-продукту.

Внаслідок аналізу результатів тестування, було визначено деякі зміни для покращення продукту, а саме: додавання мапи для збору даних від користувача та необхідність позбутись суб'єктивності оцінювання експертом, що можна зробити або запровадженням колективного експерту, або повністю автоматичним проставленням необхідних даних для розрахунку оцінки з використанням зібраної заздалегідь бази даних та інформації з зовнішніх ресурсів. Більше того, в наступній версії продукту планується врахувати фактори ризику та часу, що зробить систему ще більш унікальною.

Щодо використання даного продукту, то дану систему можна використати для прогнозування оціночної вартості ОН, з метою забезпечення розумної інвестиції заощаджень та попереднього аналізу всіх ризиків. Даний продукт буде корисним та необхідним в першу чергу потенційному інвестору, який хоче зробити будь-які маніпуляції з об'єктами нерухомості, а також для компаній, що займаються продажем і викупом нерухомості.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Інвестиції в нерухомість — плюси і мінуси інвестування [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – Вересень 23, 2020. – Режим доступу: <https://vseznayko.com.ua/investicii-v-neruhomist-pljusi-i-mi.html>
2. Плюси і мінуси володіння будинком в Америці [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – О. Українка – Квітень 3, 2022. – Режим доступу: <https://www.forumdaily.com/uk/plyusy-i-minusy-vladeniya-domom-v-amerike/>
3. Financial Institutions Reform, Recovery, and Enforcement Act (FIRREA) [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – Ю. Каган – Травень 26, 2022. – Режим доступу: <https://www.investopedia.com/terms/f/financial-institutions-reform-recovery-enforcement.asp>
4. Dodd-Frank Wall Street Reform and Consumer Protection Act [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – Т. Темплін – Квітень 24, 2023. – Режим доступу: https://www.financestrategists.com/wealth-management/dodd-frank/?gclid=Cj0KCQjwyLGjBhDKARIsAFRNgW_Hi55gxQ-jHqDEXsUGZi2VFpdz0CyKsynfXv55XzdIW4WE2m_XWJoaAo_2EALw_wcB
5. What is USPAP [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – Грудень 31, 2023. – Режим доступу: https://www.appraisalfoundation.org/imis/TAF/Standards/Appraisal_Standards/Uniform_Standards_of_Professional_Appraisal_Practice/TAF/USPAP.aspx?hkey=a6420a67-dbfa-41b3-9878-fac35923d2af
6. Raymond J. W. Real Estate Appraisal: Principles and Procedures: підручник / Raymond J. W. – Educational Textbook Company, 2019 – ISBN978-1626843639.
7. Нерухомість США [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – Режим доступу: <https://xn--80atid4b5f.xn-->

j1amh/%D0%BD%D0%B5%D1%80%D1%83%D1%85%D0%BE%D0%BC%D1%96%D1%81%D1%82%D1%8C-%D1%81%D1%88%D0%B0/

8. 8 critical factors that influence a home's value [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – Joe Gomez – Липень 4, 2022. – Режим доступу: <https://www.opendoor.com/articles/factors-that-influence-home-value>
9. Практичні питання оцінки об'єктів нерухомості [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – С. М. КОБЗАН – 2017. – Режим доступу: <https://core.ac.uk/download/pdf/147455411.pdf>
10. Оцінка нерухомості витратним підходом [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – О. Івченко – Липень 9, 2020. – Режим доступу: <https://pareto.com.ua/ua/blog/ots-nka-nerukhomost-vitratnim-p-dkhodom/>
11. Сучасний міжнародний досвід впровадження інформаційних технологій в галузі аналізу ринку та оцінки нерухомості [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – 2017. – Режим доступу: <https://www.uvecon.ua/ua/suchasniy-mizhnarodniy-dosvid-vprovadzhennya-informatsiynih.html>
12. Zillow [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. . – 2006 : 2023 р. – <https://www.zillow.com/>
13. What is a Zestimate? [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – <https://www.zillow.com/z/zestimate/>
14. What is a Redfin? [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – <https://www.redfin.com/>
15. Realtor.com [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – <https://www.realtor.com/>
16. Регресійний аналіз [Електронний ресурс]: електронні ресурси в науці, культурі та освіті. – <http://epi.cc.ua/282-regressionnyiy-analiz-26451.html>
17. Сутність поняття ринкова вартість [Електронний ресурс] : [Веб-сайт] – Електр. дані. – Режим доступу: <http://www.afo.com.ua/uk/news/2/1390>

- 18.І. Ніковський «Багатокритеріальна оптимізація» [Електронний ресурс] : [Навчальний посібник] – 2016 – Електр. дані. – Режим доступу: [https://sau.nmu.org.ua/ua/osvita/metod/Models_and_decision-making_techniques\(Us_Koryashkina\)_NMU_SAU.pdf](https://sau.nmu.org.ua/ua/osvita/metod/Models_and_decision-making_techniques(Us_Koryashkina)_NMU_SAU.pdf)
- 19.Задача багатокритеріальної оптимізації [Електронний ресурс] : [Веб-сайт] – Електр. дані. – Режим доступу: <http://ebib.pp.ua/zadachi-mnogokriterialnoy-optimizatsii-7818.html>
- 20.Saaty T. L. (2005). Models, Methods, Concepts & Applications of the Analytic Hierarchy Process. Springer Science+Business Media. Режим доступу: https://www.google.com.ua/books/edition/Models_Methods_Concepts_Applications_of/6J9XI8I1qjwC?hl=uk&gbpv=0
- 21.ISAHP [Електронний ресурс] : [Веб-сайт] – Електр. дані – Режим доступу: https://en.wikipedia.org/wiki/International_Symposium_on_the_Analytic_Hierarchy_Process
- 22.Sultana N. & Sultana S. (2019). Using Artificial Intelligence in Real Estate Risk Management. International Journal of Computer Science and Technology, 10(3), 174-178. doi: 10.24297/ijct.v10i3.8106
- 23.Sharma N. Predicting Housing Prices with Machine Learning using Python: підручник / Sharma N. and Dhull P., 2020 рік. Режим доступу: <https://towardsdatascience.com/predicting-housing-prices-with-machine-learning-using-python-545c348082e5>
- 24.Chen, J., Lu, M., Liu, Y., & Wang, Q. (2018). Forecasting Real Estate Prices with Machine Learning Techniques. International Journal of Strategic Property Management, 22(1), 1-14. doi: 10.3846/ijspm.2018.5406

Додаток А

Опис процесів автентифікації

Процеси автентифікації — це процеси перевірки ідентичності користувача перед наданням доступу до розробленої системи. Тож, розглянемо процеси автентифікації нашої системи — реєстрацію та логін. Розпочнемо з реєстрації, відповідна сторінка, окрім форми для реєстрації, містить ще перехід на сторінку логіна, якщо користувач уже зареєстрований в системі (рисунок 4.1).

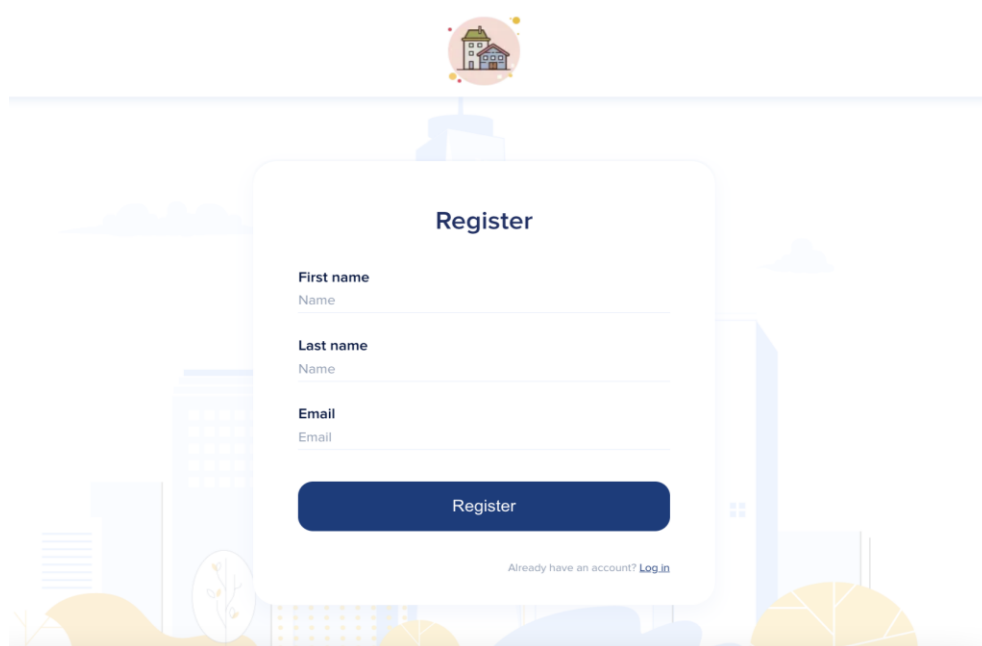


Рисунок 4.1 — Сторінка для реєстрації

Форма реєстрації містить базову перевірку на валідність введених даних, результати при неправильному вводі представлено на рисунках 4.2 – 4.3.

Register	Register
First name Inna	First name Inna
Last name Grynishak	*Last name Name Required
*Email ingrynishak Invalid email	Email ingrynishak@grandz.io
Register	Register

Рисунки 4.2 – 4.3 — Приклади помилок при неправильно заповненій формі

У випадку коректно заповненої форми, при натискання на кнопку «Register», з'являється сторінка результату реєстрації з повідомленням про потребу перевірити пошту(рисунки 4.4), куди повинен прийти лист з подальшою інструкцією (рисунки 4.5).

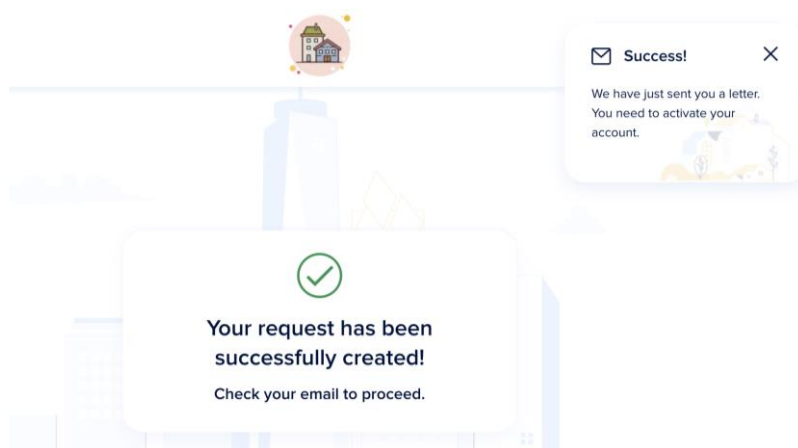


Рисунок 4.4 — Результат реєстрації

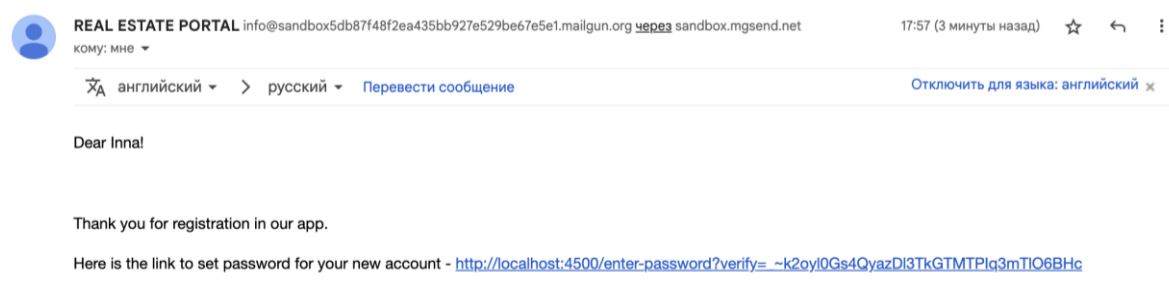


Рисунок 4.5 — Лист з інструкцією для закінчення процесу реєстрації

Лист з рисунку 4.5 містить посилання на сторінку з формою створення паролю користувача (рисунки 4.6).

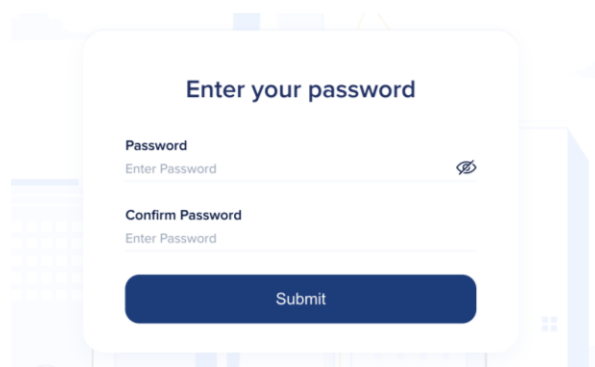


Рисунок 4.6 — Форма для створення паролю користувача

Дана форма містить базову перевірку введених даних(рисунки 4.7 – 4.8).

The image shows two side-by-side screenshots of a web form titled "Enter your password".

Left Screenshot (Figure 4.7): The "Password" field contains "Enter Password" and has a red error message below it: "Must have at least 8 characters". The "Confirm Password" field also contains "Enter Password". A blue "Submit" button is at the bottom.

Right Screenshot (Figure 4.8): The "Password" field contains "....." and the "Confirm Password" field contains ".....". A red error message below the confirm field reads: "Passwords do not match". A blue "Submit" button is at the bottom.

Рисунки 4.7 – 4.8 — Приклади помилок при неправильно заповненій формі

При успішному встановленню паролю система перенаправляє користувача на сторінку логіну (рисунок 4.9).

The image shows a "Log In" form centered on a light blue background with a cityscape illustration.

The form contains the following elements:

- Title:** "Log In"
- User name:** A text input field with the placeholder "Enter Email".
- Password:** A text input field with the placeholder "Enter Password" and a small eye icon for toggling visibility.
- Remember me:** A checkbox with the label "Remember me".
- Log In:** A prominent blue button.
- Links:** "Forgot password" and "Not registered? Register now" are located at the bottom of the form.

Рисунок 4.9 — Сторінка логіну в систему

На цьому моменті, процеси автентифікації завершені і користувач має змогу працювати з ресурсами системи для оцінки об'єктів нерухомості. Після завершення огляду процесів автентифікації можна зробити висновок, що автентифікація є дуже важливим етапом для захисту від несанкціонованого доступу до систем та даних користувачів.

Додаток Б

Основні програмні модулі для оцінки

```

import { Router } from 'express'
import { StatusCodes } from 'http-status-codes'

import { addAsync } from '@awaitjs/express'
import * as env from '@shared/env'
import { bufferFromGCS } from '@services/cloudStorage'

import User from '@models/user'
import TransactionLog from '@models/transactionLog'
import Document, { DocumentType } from '@models/document'
import Transaction, { TransactionStatus } from '@models/transaction'
import Deal, { ConditionType, DealStatus } from '@models/deal';
import Notification, { NotificationType } from '@models/notification';
import { getMailgunClient } from '@clients/mailgun';

const router = addAsync(Router())

// API endpoint to calculate the final state value
router.postAsync('/deals/:id', async (req, res) => {
  const dealId = req.params.id
  const weights = req.body.weights
  const toMetro = req.body.toMetro
  const toCenter = req.body.toCenter
  const toSupermarket = req.body.toSupermarket
  const city = req.body.city
  const state = req.body.state
  const error = req.body.error
  const change = req.body.change

  const deal = await Deal.findByPk(dealId)
  if (!deal) {
    return res.status(StatusCodes.BAD_REQUEST).json({
      error: 'Deal was not found',
    })
  }

  if (error || !(toMetro && toCenter && toSupermarket && city && state)) {
    await Notification.create({ dealId: deal.id, type: NotificationType.RequestDeclined })
    await deal.update({ status: DealStatus.Declined })
    return res.status(StatusCodes.OK).json({ msg: 'End' })
  }

  const balcony = deal.balcony ? 1 : 0
  const getValue = (type: ConditionType) => {
    switch (type) {
      case ConditionType.Terrible:
        return 2
      case ConditionType.Poor:

```

```

    return 4
  case ConditionType.Average:
    return 6
  case ConditionType.Good:
    return 8
  case ConditionType.Excellent:
    return 10
  }
}

const mailgunClient = getMailgunClient()
try {
  const amount = change * (
    deal.area * weights.area +
    balcony * weights.balcony +
    deal.roomCount * weights.roomCount +
    deal.floor * weights.floor +
    city * weights.city +
    state * weights.state +
    toSupermarket * weights.toSupermarket +
    toMetro * weights.toMetro +
    toCenter * weights.toCenter +
    getValue(deal.condition) * weights.condition +
    getValue(deal.buildCondition) * weights.buildCondition
  )

  await Notification.create({ dealId: deal.id, type: NotificationType.RequestCompleted })

  mailgunClient.sendSuccessFileUploadLetter({ 'v:userName': 'Inna', 'v:fileUrl': '' })

  await deal.update({ amountToSell: amount, status: DealStatus.Completed })

  return res.status(StatusCodes.OK).json({
    msg: 'Value of real estate is calculated successfully',
  })
} catch (error) {
  // eslint-disable-next-line no-console
  console.error(`Error while calculating value of real estate: ${error}`)
  return res.status(StatusCodes.INTERNAL_SERVER_ERROR).json({
    error: 'Internal server error',
  })
}
})

// API endpoint to get information
import { Router } from 'express'

import { addAsync } from '@awaitjs/express'
import { StatusCodes } from 'http-status-codes'
import User from '@models/user'
import Deal, { DealStatus } from '@models/deal'

```

```

import { isPositiveNumber } from '@shared/validators'
import { getCommissionPrice } from '@utils/getCommissionPrice'
import { authenticateUser } from '@middlewares/authentication'
import Notification, { NotificationType } from '@models/notification'

import {
  startCreateDocuSignEnvelopeForDeal,
  startCreateSalesforceOpportunity,
  startSendDealCreatedEmail,
} from '@shared/bull/bull-client'

const router = addAsync(Router())

const requiredDealFields = [
  'brokerage',
  'area',
  'condition',
  'buildCondition',
  'roomCount',
  'floor',
  'balcony',
  'latitude',
  'longitude',
  'street',
  'city',
  'state',
  'dateOfCommission',
]

router.postAsync('/', authenticateUser, async (req, res) => {
  let fields = new Map()
  console.log(req.body)
  const missingFields: string[] = []
  requiredDealFields.forEach(fieldName => {
    if (req.body[fieldName]) {
      fields.set(fieldName, req.body[fieldName])
    } else {
      missingFields.push(fieldName)
    }
  })

  if (missingFields.length) {
    return res.status(StatusCodes.BAD_REQUEST).json({
      error: `Required fields are missing: ${missingFields.join(', ')}`,
    })
  }

  const user: User = res.locals.currentUser

  let deal: Deal
  try {
    fields.set('userId', user.id).set('status', DealStatus.Pending).set('dateOfFunding', new Date())
  }

```

```

fields.set('data', { commission: '...' })
fields = Object.fromEntries(fields)

deal = await Deal.create({ ...fields })

await Notification.create({
  dealId: deal.id,
  type: NotificationType.RequestCreated,
})
} catch (error) {
  // eslint-disable-next-line no-console
  console.error(`Error while creating deal: ${error}`)
  return res.status(StatusCodes.INTERNAL_SERVER_ERROR).json({
    error: 'Internal server error',
  })
}

return res.status(StatusCodes.CREATED).json({
  msg: 'Request for deal creation successful',
  deal: deal,
})
})

function calculateFinalScore(criteriaValues: number[], weights: number[]): number {
  const n = criteriaValues.length;
  let finalScore = 0;

  for (let i = 0; i < n; i++) {
    finalScore += criteriaValues[i] * weights[i];
  }

  return finalScore;
}

// functions for calculate criteria weights
function calculateWeights(matrix: number[][]): number[] {
  const n = matrix.length;
  const weights: number[] = [];

  // Calculate sum of rows
  const rowSum: number[] = matrix.map(row => row.reduce((acc, val) => acc + val, 0));

  const normalizedMatrix: number[][] = matrix.map(row =>
    row.map((value, colIndex) => value / rowSum[colIndex])
  );

  // Calculate weights
  for (let i = 0; i < n; i++) {
    const rowSumNormalized = normalizedMatrix[i].reduce((acc, val) => acc + val, 0);
    const weight = rowSumNormalized / n;

```

```

    weights.push(weight);
  }

  return weights;
}

// functions for check on consistency
function calculateConsistency(matrix: number[][]): number {
  const n = matrix.length;
  const rowSum: number[] = [];
  const weightedSum: number[] = [];

  for (let i = 0; i < n; i++) {
    let sum = 0;
    for (let j = 0; j < n; j++) {
      sum += matrix[i][j];
    }
    rowSum.push(sum);
  }

  for (let i = 0; i < n; i++) {
    let sum = 0;
    for (let j = 0; j < n; j++) {
      sum += matrix[i][j] / rowSum[j];
    }
    weightedSum.push(sum);
  }

  // Calculate узгодженості (CI)
  const CI = (Math.max(...weightedSum) - n) / (n - 1);

  // Table values (RI)
  const randomIndex: number[] = [0, 0, 0.58, 0.9, 1.12, 1.24, 1.32, 1.41, 1.45, 1.49];

  // Обчислення відношення узгодженості (CR)
  const CR = CI / randomIndex[n - 1];

  if (CR > 0.1) {
    console.log("Матриця попарних порівнянь не є узгодженою.");
  } else {
    console.log("Матриця попарних порівнянь є узгодженою.");
  }

  return CR;
}

```