

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві _____ Белевцю Роману Валерійовичу
(прізвище, ім'я, по батькові)1. Тема роботи Розроблення інтелектуального модулю для моніторингу цін товарів з паперового чеку та впровадження його у мобільний застосунок для відстеження витрат

затверджена наказом університету від 26 травня 2026 року № 435Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 19 червня 2026 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, мова програмування Python, бібліотека Google GenAI, бібліотека OpenAI, бібліотека Anthropic, бібліотека Pydantic, великі мовні моделі, сервіс вебпошуку Tavily, сервіс завантаження сторінок Firecrawl, картографічна база OpenStreetMap, середовище прототипування Figma.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз існуючих сервісів порівняння цін та підходів до автоматизованого пошуку цін на основі вебскрейпінгу та великих мовних моделей.2. Формування набору даних касових чеків та визначення метрик оцінки якості.3. Розробка та експериментальне порівняння чотирьох підходів до моніторингу цін, включаючи порівняльний аналіз LLM різних розробників.4. Проектування архітектури інтелектуального модуля на основі обраного підходу.5. Розробка модуля та впровадження у мобільний застосунок Billka AI.6. Оцінка якості роботи модуля на незалежному контрольному наборі.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми моніторингу цін та огляд існуючих рішень, постановка задачі, порівняння чотирьох підходів до пошуку цін (вебскрейпінг, послідовність LLM, простий агент, гібридний агент), архітектура гібридного агента, ілюстрація роботи модуля на прикладі товару з чека, результати експериментального порівняння підходів, впровадження модуля у мобільний застосунок Billka AI, оцінка якості на контрольному наборі.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	13.04.2026	
2	Аналіз завдання, підбір літератури	14.04.26-16.04.26	
3	Аналіз літератури з проблеми, що вирішується	17.04.26-19.04.26	
4	Аналіз існуючих методів розпізнавання	20.04.26-22.04.26	
5	Формування кроків вирішення проблеми	23.04.26-05.05.26	
6	Програмна реалізація	26.04.26-20.05.26	
7	Аналіз отриманих результатів	21.05.26-04.06.26	
8	Оформлення пояснювальної записки	05.06.26-09.06.26	
9	Перевірка на нормоконтроль	10.06.26-19.06.26	
10	Перевірка на плагіат	11.06.26-30.06.26	
11	Рецензування	20.06.26-30.06.26	
12	Підготовка презентації та доповіді	20.06.26-30.06.26	
13	Занесення роботи в електронний архів	30.06.26-09.07.26	
14	Попередній захист кваліфікаційної роботи	30.06.26-09.07.26	

Дата видачі завдання 13 квітня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Яковлева О. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 132 с., 19 табл., 23 рис., 9 дод., 59 джерел.

АГЕНТ, ВЕБСКРЕЙПІНГ, ВЕЛИКА МОВНА МОДЕЛЬ, КАСОВИЙ ЧЕК, МОНІТОРИНГ ЦІН, OPENSTREETMAP, PYTHON.

Об'єктом роботи є процес моніторингу та порівняння цін на товари з різних торговельних мереж на основі інформації з касових чеків.

Метою роботи є розроблення інтелектуального модулю для моніторингу цін товарів з паперового чеку та впровадження його у мобільний застосунок для відстеження витрат.

Під час роботи використано методи штучного інтелекту, інформаційного пошуку, обробки відкритих картографічних даних OpenStreetMap.

Розроблено інтелектуальний модуль моніторингу цін, що автоматизує порівняння цін на товари з паперового чеку серед торговельних мереж міста користувача, проведено порівняння чотирьох підходів та обрано оптимальний.

Область застосування: мобільні застосунки для відстеження витрат, вебзастосунки для моніторингу цін, аналіз споживчого кошика.

Результати дослідження отримані в межах міжнародного проєкту INITIATE за грантом No. 101136775 HORIZON WIDERA 2023 ACCESS 03.

AGENT, WEB SCRAPING, LARGE LANGUAGE MODEL, RECEIPT, PRICE MONITORING, OPENSTREETMAP, PYTHON.

The objective of this work is the process of monitoring and comparing prices of products from different retail chains based on information from cash receipts.

The goal of this work is to develop an intelligent module for monitoring prices of products from a paper receipt and to integrate it into a mobile application for expense tracking.

The following were used during the work: artificial intelligence methods, information retrieval methods, and open cartographic data processing methods based on OpenStreetMap

An intelligent price monitoring module has been developed that automates price comparison for products from a paper receipt across retail chains in the user's city; a comparison of four approaches has been conducted, and the optimal one has been selected.

Applications: mobile applications for expense tracking, web applications for price monitoring, consumer basket analysis.

The results of this research were obtained under the international project INITIATE under the grant No. 101136775 HORIZON WIDERA 2023 ACCESS 03.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ.....	9
1 Огляд поточного стану питання моніторингу та порівняння цін на товари.....	12
1.1 Постановка проблеми пошуку ідентичних товарів у різних торговельних мережах.....	12
1.2 Аналіз можливостей поточних рішень	14
1.3 Можливість моніторингу цін у застосунках для відстеження витрат на основі чеків.....	18
1.4 Підходи, методи та інструменти для вирішення задачі порівняння цін.....	20
1.5 Постановка задачі	23
2 Розробка методу моніторингу цін на товари на основі інформації з касових чеків	25
2.1 Постановка завдання та критерії оцінки якості.....	25
2.2 Створення власного набору даних чеків з ідентичними товарами з різних торговельних мереж	26
2.3 Підхід на основі вебскрейпінгу з браузерами без графічного інтерфейсу.....	28
2.3.1 Архітектура та компоненти підходу	29
2.3.2 Експериментальна перевірка	29
2.3.3 Висновки щодо точності, часових витрат та вартості.....	31
2.4 Підхід на основі чіткої послідовності великих мовних моделей...	32
2.4.1 Архітектура послідовності та її компоненти	33
2.4.2 Принципи відбору моделей для експериментів.....	34
2.4.3 Зміст компонентів послідовності	34
2.4.4 Експерименти з вибору моделей	34
2.4.5 Експериментальна перевірка послідовності	35

	6
2.4.6	Висновки щодо точності, часових витрат та вартості..... 37
2.5	Підхід на основі інтелектуального агента..... 38
2.5.1	Концепція агентного підходу 39
2.5.2	Архітектура агента..... 39
2.5.3	Доступні інструменти агента 40
2.5.4	Експерименти з вибору моделі агента 41
2.5.5	Приклад роботи агента на типовому запиті 42
2.5.6	Експериментальна перевірка агента 43
2.5.7	Висновки щодо точності, часових витрат та вартості..... 44
2.6	Підхід на основі гібридного агента над послідовністю LLM 46
2.6.1	Концепція гібридного підходу..... 46
2.6.2	Інструменти агента..... 46
2.6.3	Робочий цикл агента 48
2.6.4	Експерименти з вибору моделі агента-оркестратора 48
2.6.5	Експериментальна перевірка підходу 49
2.6.6	Висновки щодо точності, часових витрат та вартості..... 51
2.7	Розробка алгоритму моніторингу цін товарів з чеків 52
2.7.1	Порівняння розглянутих підходів 52
2.7.2	Алгоритм моніторингу цін на основі гібридного агента 54
3	Розробка мобільного застосунку моніторингу цін на товари
	з паперового чека 56
3.1	Технічне завдання 56
3.2	Налаштування програмного середовища 58
3.3	Особливості підключення та налаштування LLM та зовнішніх сервісів 59
3.4	Розробка архітектури модуля 61
3.5	Опис мобільного застосунку Billka AI та можливості впровадження модуля..... 65
3.6	Розробка модуля моніторингу цін..... 67
3.7	Впровадження модуля пошуку цін у застосунок Billka AI 72

	7
3.8 Ілюстрація роботи модуля	74
3.9 Оцінка якості роботи модуля.....	78
Висновки	81
Перелік джерел посилан	84
Додаток А Експериментальний набір даних	90
Додаток Б Архітектурні схеми підходів	98
Додаток В Результати наскрізного тестування за окремими позиціями.....	100
Додаток Г Скріншоти інтерфейсу arena ai.....	106
Додаток Д Результати з вибору моделей	110
Додаток Е Опис етапів послідовності LLM	122
Додаток Ж Перелік моделей для роботи	127
Додаток И Результати з вибору моделей-оркестраторів для агентів.....	129
Додаток К Приклад роботи інтелектуального агента	132

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

адапт. – адаптивний

англ. – англійською мовою

м. – місто

фікс. – фіксований

США – Сполучені Штати Америки

API – Application Programming Interface (інтерфейс програмного застосунку)

AVG_w – зважене середнє коефіцієнтів Дайса та Жаккара як метрик подібності множин

F1-міра – гармонійне середнє між точністю та повнотою

JSON – JavaScript Object Notation (нотація об'єктів JavaScript, формат обміну даними)

LLM – Large Language Model (велика мовна модель)

MLLMs – Multimodal Large Language Models (мультимодальні великі мовні моделі)

ReAct – Reasoning and Acting (методологія агентного циклу «міркування, дія, спостереження»)

URL – Uniform Resource Locator (уніфікований локатор ресурсу)

ВСТУП

Моніторинг та порівняння цін на товари роздрібної торгівлі розділяється за трьома основними напрямками: збирання інформації про ціни з вебсайтів торговельних мереж, інтелектуальна інтерпретація скорочених назв товарів та зіставлення товарних позицій між різними джерелами.

Збирання інформації про ціни – це автоматичне отримання даних про товари і їхню вартість з вебсайтів торговельних мереж за допомогою спеціалізованих програмних інструментів. Цей напрямок історично реалізується методами вебскрейпінгу через браузері без графічного інтерфейсу та має певні обмеження: вразливість до зміни структури вебсторінок, залежність від наявності у магазину власного вебсайту і недостатню точність роботи зі скороченими назвами.

Інтерпретація скорочених назв товарів є важливою підзадачею, специфічною для роботи з паперовими касовими чеками. Жорсткі обмеження довжини друкованого рядка чеку, які зазвичай становлять від 10 до 25 символів, змушують торговельні мережі використовувати локальні скорочення, аббревіатури та внутрішні позначення, які не є стандартизованими та не документуються у відкритому доступі. Наприклад, безлактозне молоко Raјo об'ємом 1 літр на чеку мережі Lidl у Словаччині записано як «Lktf. mlieko», а в мережі COOP Jednota – як «Mlieko raјo bezl 1,5 % 1l»; яблука сорту Gala на чеку австрійської мережі BILLA записано як «DKIH Apfel "Gala" gelegt», а в мережі SPAR – як «APFEL GALA». Розв'язання задачі зіставлення таких записів стало можливим завдяки розвитку великих мовних моделей (LLM), здатних інтерпретувати скорочення з урахуванням мовного та доменного контексту.

Не зважаючи на суттєвий прорив в області комп'ютерного зору [1], наприклад, в задачах підрахунку людей [2, 3], розпізнавання облич [4, 5], аналізу текстур [6], екстракції текстової інформації із зображень [7] та інших, назви, які розпізнаються з фото паперового чека можуть мати незначні

помилки, що створює додаткові складнощі. Ці помилки обумовлені якістю самого чека та залежать від методу розпізнавання. Для розпізнавання тексту на зображенні можна використовувати підхід на основі технології оптичного розпізнавання символів (англ. Optical Character Recognition, OCR), де зображення потребують передобробки, а саме бінарізації, покращення контрастності, компенсації геометричних перетворень. Як один з найпоширеніших методів для компенсації геометричних перетворень застосовуються методи на базі дескрипторів [8–13]. Також можна використовувати більш сучасний підхід на основі MLLMs [14–17].

Якість розпізнавання залежить від якості самого фото та також від метода, що використовується.

Зіставлення товарних позицій між різними джерелами полягає в автоматичному встановленні відповідності між описами одного й того самого товару у різних торговельних мережах та платформах електронної комерції. Цей напрямок є основою для побудови сервісів порівняння цін. Області застосування – мобільні застосунки для відстеження витрат, сервіси порівняння цін, аналіз споживчого кошика, статистичний аналіз вартості споживчого кошика та інфляції.

Актуальність роботи полягає у потребі розроблення універсального модульного рішення, придатного для інтеграції у мобільні та вебзастосунки, яке автоматизує пошук і порівняння цін на товари з паперового касового чека на основі сучасних технологій штучного інтелекту, без прив'язки до конкретних торговельних мереж та з можливістю роботи з довільною країною та мовою. Наявні сервіси порівняння цін працюють за моделлю прямих інтеграцій з кожною мережею, що обмежує їх географічне покриття та фіксує перелік партнерів; застосунки для відстеження витрат використовують дані з чеків лише для обліку, а не для пошуку цін на ті самі товари в інших мережах. Заповнення цієї прогалини відкриває нові можливості як для кінцевих споживачів, які зацікавлені у виборі найвигідніших пропозицій, так і для бізнес-аналітики та статистичних служб.

Сучасний стан розвитку технологій штучного інтелекту дозволяє розглянути кілька принципово різних підходів до розв’язання поставленої задачі. Перший – класичний вебскрейпінг через браузері без графічного інтерфейсу, що відображає ранній етап розвитку галузі і не використовує LLM. Другий підхід реалізує чітку послідовність LLM з вбудованим інструментом вебпошуку, де кожна модель виконує окрему спеціалізовану задачу обробки тексту та видобування інформації. Третій підхід базується на інтелектуальному агенті, побудованому навколо однієї LLM, що самостійно обирає послідовність дій залежно від проміжних результатів. Експериментальне порівняння цих підходів за метриками точності, охоплення магазинів, часу обробки та вартості одного запиту дозволяє обґрунтовано обрати оптимальний для практичної реалізації.

Дана робота присвячена розробленню інтелектуального модуля для моніторингу цін товарів з паперового касового чеку та впровадженню його у мобільний застосунок для відстеження витрат.

1 ОГЛЯД ПОТОЧНОГО СТАНУ ПИТАННЯ МОНІТОРИНГУ ТА ПОРІВНЯННЯ ЦІН НА ТОВАРИ

1.1 Постановка проблеми пошуку ідентичних товарів у різних торговельних мережах

Сучасний ринок роздрібної торгівлі характеризується значною кількістю торговельних мереж, платформ електронної комерції та онлайн-каталогів, кожен з яких формує власну систему найменувань товарів. Ця система відображає внутрішні особливості упорядкування товарів, маркетингові рішення та технічні обмеження інформаційних систем конкретної мережі. Як наслідок, один і той самий товар може мати десятки різних варіантів опису в різних джерелах. На перший погляд, задача встановлення відповідності між двома записами здається тривіальною. Проте відсутність єдиного стандарту найменувань перетворює її на складну проблему, відому в науковій літературі як зіставлення товарів (англ. product matching) [18].

Розбіжності між описами одного й того ж товару проявляються на кількох рівнях. На лексичному рівні спостерігаються відмінності у порядку слів, використанні синонімів та різному розташуванні бренду (наприклад, «Молоко пастеризоване Яготинське 2,5 % 900 г» та «Яготинське молоко пастеризоване жирність 2,5 % 900 г» оп'исують той самий товар). На орфографічному рівні – наявність або відсутність діакритичних знаків, транслітерація іншомовних назв («Coca-Cola» та «Кока-Кола»), різні формати запису одиниць вимірювання («0,5 л», «500 мл» та «0,5L»). На структурному рівні – пропуск, перестановка або різне оформлення технічних атрибутів товару (об'єм, маса, вміст жиру, потужність, об'єм пам'яті). Наявність таких розбіжностей унеможливорює просте текстове порівняння назв і потребує інтелектуальної нормалізації даних перед порівнянням.

Описана проблема є типовою для багатьох категорій роздрібно́ї торгівлі: побутової техніки, одягу, автозапчастин, фармацевтичних препаратів, музичних інструментів, книжкової продукції, а також харчових товарів. Наприклад, одна й та сама модель холодильника у каталогах різних мереж може бути подана з різним порядком атрибутів, у різному форматі запису об'єму або з різною повнотою характеристик. Таким чином, задача зіставлення товарів має міжгалузевий характер і потребує розроблення універсальних підходів, стійких до варіативності опису. Особливий практичний інтерес становить категорія продуктів харчування через високу частоту покупок, значну кількість торговельних мереж у цій галузі та регулярну зміну цінових пропозицій.

Практична цінність розв'язання цієї задачі визначається кількома основними сценаріями використання. Першим є моніторинг цін у динаміці – відстеження того, як змінюється вартість одного й того самого товару в межах однієї або кількох торговельних мереж протягом тривалого часу. Другим сценарієм є порівняння цін у поточному моменті часу, що дає споживачеві змогу прийняти обґрунтоване рішення щодо місця придбання конкретного товару. Третій сценарій полягає у побудові спеціалізованих сервісів порівняння цін, які об'єднують пропозиції різних продавців у межах вузької категорії, наприклад, лише продукти харчування, лише медикаменти або лише побутова техніка певного класу.

Крім безпосередньо комерційного застосування, задача зіставлення товарів має також статистичне значення: державні служби статистики використовують результати зіставлення для розрахунку показників інфляції та моніторингу вартості споживчого кошика. Проте, незважаючи на широке коло застосувань, задача автоматизованого пошуку ідентичних товарів у різних торговельних мережах залишається складною через відсутність єдиного стандарту найменувань та значну варіативність опису одних і тих самих товарів.

1.2 Аналіз можливостей поточних рішень

З огляду на описану проблему на ринку сформувався окремий клас програмних рішень, сервіси порівняння цін, що збирають дані про товари і ціни з різних торговельних мереж, зводять їх до єдиної структури та надають кінцевому користувачеві інтерфейс для пошуку і порівняння. Наявні рішення розрізняються за географією (локальні або міжнародні), форматом доступу (вебсайт, мобільний застосунок або комбіноване рішення) та спеціалізацією (сервіси широкого профілю або спеціалізовані на конкретній категорії товарів).

Найбільшим глобальним сервісом порівняння цін є «Google Shopping» [19], інтегрований у пошукову систему Google. Сервіс об'єднує пропозиції з онлайн-магазинів переважній більшості країн світу. На європейському ринку діють «Idealo» [20] – німецький сервіс, що функціонує у шести країнах Європи (Німеччина, Великобританія, Франція, Італія, Іспанія, Австрія) та збирає пропозиції з понад 50 000 онлайн-магазинів, «PriceRunner» [21] – шведський сервіс, що працює у Швеції, Данії, Норвегії та Великобританії, а також «Hotline.ua» [22] – найбільший сервіс порівняння цін в Україні з охопленням понад 1 000 онлайн-магазинів. Зазначені сервіси охоплюють переважно побутову електроніку, техніку, одяг, косметику та інші непродовольчі категорії. У цих категоріях порівняння спрощується завдяки стандартизованим артикулам та міжнародним штриховим кодам. Водночас товари вузьких категорій, таких як продукти харчування, фармацевтичні препарати та автозапчастини, на цих платформах представлені обмежено або відсутні, що зумовлює потребу у спеціалізованих рішеннях.

Спеціалізовані сервіси для конкретних галузей значно менш поширені та здебільшого мають локальний характер. У сегменті продуктів харчування функціонують «Де Дешевше», «GoToShop.ua» (Україна), «Priceo.sk», «Kompaszliav.sk», «Kdejeakcia.sk» (Словаччина), «Kassalapp», «Enhver»

(Норвегія). У галузі фармацевтики на українському ринку представлені «Tabletki.ua» [23], «Apteki.ua» [24] та «Liki.ua» [25] – сервіси, що вирішують аналогічну задачу зіставлення, оскільки одні й ті самі лікарські засоби можуть продаватися під різними торговельними назвами та в різному дозуванні. Фармацевтичні сервіси працюють за тим самим принципом прямих інтеграцій з аптечними мережами і мають ті самі обмеження – локальну прив’язку та фіксований перелік партнерів.

Сервіс «Де Дешевше» [26] – українське рішення у форматі мобільного застосунку для iOS та Android, до якого підключено 21 торговельну мережу (серед яких «АТБ», «Сільпо», «Фора», «Метро», «Ашан», «Varus», «Novus», «Рукавичка» та інші). Можливості застосунку охоплюють пошук товарів за назвою або брендом, сканування штрихкодів для порівняння цін у різних мережах, калькулятор ціни за одиницю вимірювання, зберігання дисконтних карток, відстеження зміни цін на обрані товари та індивідуальні (персоналізовані) рекомендації на основі алгоритмів штучного інтелекту. Суттєвим обмеженням сервісу є залежність від прямих інтеграцій з торговельними мережами. Асортимент доступних цін формується виключно з каталогів, які мережі самостійно передають до сервісу.

Сервіс «GoToShor.ua» [27] – український сервіс порівняння цін у форматі вебсайту та мобільних застосунків, до якого підключено понад 200 торговельних мереж у понад 400 містах України. Сервіс надає порівняння цін за категорією або брендом, перегляд оцифрованих акційних каталогів і листівок, додавання знижок до обраного; сканування штрихкодів не реалізовано.

Сервіс «Kassalapp» [28] – норвезьке рішення у форматі мобільного застосунку та вебінтерфейсу, що охоплює понад 100 000 товарів. Сервіс є найбільш функціональним серед розглянутих: сканування штрихкодів із порівнянням цін, відстеження динаміки цін, мапа понад 4 000 магазинів, інтеграція з банківськими рахунками для обліку витрат та сканування паперових касових чеків для фіксації покупок. Слід зазначити, що

сканування чеків у «Kassalapp» використовується для обліку витрат, а не для пошуку цін на ті самі товари в інших мережах. На норвезькому ринку також функціонує вебсервіс «Enhver» [29], що порівнює ціни з акційних каталогів різних торговельних мереж та відстежує цінові тренди на продукти харчування.

Серед словацьких рішень варто виділити «Priceo.sk» [30] – спеціалізований сервіс порівняння цін, який моніторить понад 2 000 товарів. Також на ринку представлений «Heureka.sk» [31] – міжнародний сервіс широкого профілю, що працює у дев'яти країнах Центральної та Східної Європи. «Kompasziav.sk» та «Kdejeakcia.sk» – сервіси збирання акційних листівок словацьких супермаркетів.

Зведене порівняння можливостей розглянутих сервісів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння можливостей сервісів порівняння цін

Сервіс	Формат доступу	Регіон	Спеціалізація	Сканер штрихкодів	Робота з чеками
1	2	3	4	5	6
Google Shopping	Веб + мобільний	40+ країн	Широкий профіль	-	-
Idealo	Веб + мобільний	6 країн ЄС	Широкий профіль	+	-
PriceRunner	Веб + мобільний	5 країн ЄС	Широкий профіль	-	-
Hotline.ua	Веб + мобільний	Україна	Широкий профіль	+	-
Heureka.sk	Веб + мобільний	9 країн ЦСЄ	Широкий профіль	+	-
Де дешевше	Мобільний	Україна	Продукти	+	-

Продовження таблиці 1.1

1	2	3	4	5	6
GoToShop.ua	Веб + мобільний	Україна	Продукти	-	-
Kassalapp	Веб + мобільний	Норвегія	Продукти	+	Облік витрат
Enhver	Веб	Норвегія	Продукти	-	-
Priceo.sk	Веб	Словаччина	Продукти	-	-

Дані, наведені у таблиці 1.1, дають змогу виявити кілька закономірностей. Сканування штрихкодів наявне лише у мобільних рішеннях і відсутнє у суто веборієнтованих сервісах. Стовець «Робота з касовими чеками» демонструє ключову прогалину: навіть «Kassalapp», найбільш функціональний серед розглянутих сервісів, використовує сканування чеків лише для обліку витрат, а не для пошуку цін на ідентичні товари в інших торговельних мережах на основі скорочених назв із чека.

Розглянуті сервіси мають також спільні архітектурні обмеження:

- локальна прив'язка до фіксованого переліку торговельних мереж з необхідністю індивідуальної інтеграції з кожною;
- обмежене географічне покриття без можливості масштабування між країнами;
- обмежений асортимент товарів, сформований виключно з каталогів мереж-партнерів.

Усі розглянуті рішення є кінцевими самостійними продуктами з фіксованим набором функцій. Жоден з них не пропонує можливостей для вбудовування у сторонні інформаційні системи у вигляді програмного модуля. На ринку існують модульні рішення загального призначення, наприклад, ChangeDetection.io [32]. Це відкритий інструмент мовою Python, що дає змогу моніторити довільні вебсторінки на предмет зміни вмісту

(зокрема й цін на конкретні товари в каталогах) та надсилати сповіщення про такі зміни. Проте такі рішення не вирішують задачу інтелектуального зіставлення різнойменних товарів із різних джерел. Водночас формат модуля дає змогу інтегрувати порівняння цін у застосунки суміжних напрямів, зокрема у мобільні застосунки для відстеження витрат, що працюють з касовими чеками, наприклад «Billka AI» [33, 34].

1.3 Можливість моніторингу цін у застосунках для відстеження витрат на основі чеків

Паралельно з розвитком сервісів порівняння цін на ринку мобільних застосунків активно розвивається суміжний напрям, а саме застосунки для відстеження особистих витрат на основі касових чеків. Принцип їхньої роботи полягає у розпізнаванні вмісту фотографії або сканованого зображення касового чека з використанням MLLMs, виділенні окремих товарних позицій та їхніх цін, а також у розподілі витрат за категоріями для подальшого аналізу. Такі застосунки набули популярності серед користувачів, яким важливо контролювати свої щоденні витрати на продукти харчування та товари повсякденного попиту.

Серед відомих застосунків цього класу – «Billka AI», безкоштовний мобільний застосунок для iOS та Android, що забезпечує інтелектуальне сканування касових чеків за допомогою алгоритмів штучного інтелекту, автоматичну класифікацію товарних позицій за категоріями, розділення спільних витрат між учасниками групи та візуальну аналітику витрат. Застосунок підтримує вісім мов інтерфейсу та орієнтований на повсякденне використання для контролю витрат на продукти харчування та товари повсякденного попиту. Також до цього класу належить «Splitwise» [35] – міжнародний застосунок для розділення спільних витрат між учасниками групи. У межах платної підписки застосунок підтримує сканування касових

чеків з автоматичним виділенням товарних позицій. Застосунок «Smart Receipts» [36] дає змогу сканувати чеки, формувати звіти про витрати та експортувати звітні дані. Усі перелічені застосунки використовують дані з касового чека винятково для обліку витрат, але не для пошуку цін на ті самі товари в інших торговельних мережах. Інтеграція модуля порівняння цін у такі застосунки дала б змогу не лише фіксувати витрати користувача, а й автоматично пропонувати торговельні мережі з нижчою вартістю тих самих товарів. Проте реалізація такої інтеграції ускладнюється особливостями формування назв товарів у касових чеках.

На відміну від онлайн-каталогів, де назви товарів подаються у повній формі, рядки касових чеків мають жорстке обмеження на кількість символів. Залежно від мережі та касового обладнання, на одну позицію припадає, як правило, від 10 до 25 символів. Через це торговельні мережі використовувати власні системи скорочень, аббревіатур та внутрішніх кодів для запису назв товарів. Кожна мережа формує такі скорочення за власними внутрішніми правилами, які не є стандартизованими та не документуються у відкритому доступі.

Масштаб проблеми наочно ілюструють приклади з реальних касових чеків. Один і той самий товар, безлактозне молоко Rajo 1,5 % об'ємом 1 літр, на чеку словацької мережі Lidl записано як «Lktf. mlieko». У цьому записі «Lktf.» є скороченням від маркетингового позначення «laktofree» (безлактозне), а бренд, об'єм та вміст жиру не зазначено взагалі. На чеку мережі COOP Jednota той самий товар записано як «Mlieko rajo bezl 1.5 % 1l». У цьому записі наявні бренд, скорочений атрибут «bezl» (від словацького «bezlaktózové» – безлактозне), відсоток жирності та об'єм. Аналогічна ситуація спостерігається і в інших країнах. Наприклад, на чеку австрійської мережі BILLA яблука сорту Gala записано як «DKIH Apfel "Gala" gelegt» із позначенням «DKIH». На чеку мережі SPAR той самий товар позначено просто як «APFEL GALA». Без знання контексту та специфіки скорочень конкретної мережі встановити, що такі записи стосуються одного й того

самого товару, засобами звичайного текстового порівняння неможливо.

Ця проблема має системний характер. У торговельних мережах різних країн спостерігаються аналогічні закономірності. Назви часто містять пропуск бренду або типу товару через обмеження довжини рядка, внутрішні аббревіатури, що не збігаються з загальноприйнятими скороченнями, а також запис атрибутів у довільному порядку або їхній повний пропуск. Скорочення стосуються як типу товару (наприклад, «мол. 2,5 % 1 л» замість повної назви «молоко 2,5 % жирності 1 літр»), так і всесвітньо відомих брендів (наприклад, «COKE ZERO ZERO 0,33L» на чеку мережі SPAR замість комерційної назви «Coca-Cola Zero 0,33 л»). Для категорії продуктів харчування проблема є особливо гострою. Велика різноманітність товарів з близькими назвами призводить до того, що сусідні позиції відрізняються лише окремими атрибутами, такими як вміст жиру, об'єм, наявність добавок або тип пакування.

Таким чином, порівняння цін на основі даних із касових чеків потребує попереднього етапу інтелектуальної нормалізації, тобто інтерпретації скороченого запису та визначення товару, до якого він належить. Така задача виходить за межі можливостей звичайного текстового пошуку та потребує методів, здатних аналізувати скорочення з урахуванням мовного контексту.

1.4 Підходи, методи та інструменти для вирішення задачі порівняння цін

Задача автоматизованого порівняння цін на товари з різних торговельних мереж може бути вирішена кількома концептуально різними підходами. Вибір підходу визначається насамперед тим, яким чином система отримує інформацію про ціни та як встановлює відповідність між товарами з різних джерел. Еволюція цих підходів тісно пов'язана із загальним розвитком інформаційних технологій та появою нових програмних інструментів.

До появи LLM основним підходом до автоматизованого збирання

даних про ціни був вебскрейпінг – витягування інформації з вебсторінок шляхом аналізу їхньої структури. Цей підхід передбачає завантаження вебсторінок магазинів за допомогою браузерів без графічного інтерфейсу, аналіз отриманого вмісту та виділення з нього інформації про товари і ціни. Серед програмних інструментів для реалізації цього підходу слід зазначити бібліотеку Playwright [37] для середовища Node.js, що забезпечує програмне керування браузерами без графічного інтерфейсу, та її модифікацію Patchright [38], яка забезпечує підвищену сумісність з вебсайтами, що використовують системи захисту від автоматизованого доступу. У мові Python широко застосовується бібліотека BeautifulSoup [39], призначена для аналізу отриманих вебсторінок та виділення з них структурованих даних. Для зіставлення знайдених товарів із запитом використовуються алгоритми нечіткого порівняння рядків. Зокрема, у бібліотеці RapidFuzz [40] для мови Python реалізовано відстань Левенштейна [41] і порівняння на основі біграм.

Підхід на основі вебскрейпінгу не потребує платних зовнішніх сервісів, проте має суттєві обмеження:

- вразливість до змін структури вебсторінок;
- вразливість до систем захисту від автоматизованого доступу;
- значне споживання обчислювальних ресурсів;
- низька точність зіставлення при роботі зі скороченими назвами з касових чеків.

Принципово інший підхід став можливим завдяки появі та активному розвитку LLM. Відправною точкою для масового впровадження цієї технології вважається листопад 2022 року, коли було представлено систему ChatGPT [42], що продемонструвала здатність мовних моделей розуміти контекст природної мови та працювати з неструктурованим текстом. У подальшому розвиток LLM відбувався стрімко: з'явилися моделі з можливістю виклику зовнішніх інструментів, моделі з вбудованим інструментом вебпошуку, що дає змогу здійснювати пошук в Інтернеті безпосередньо під час роботи моделі, а також моделі, здатні працювати з

різними модальностями даних, такими як текст, зображення та аудіо. Доступ до таких моделей надається через API провідних розробників, зокрема OpenAI та Google. На сьогодні доступна значна кількість моделей різних розробників, що відрізняються за якістю розв'язання різних класів задач, швидкістю роботи та вартістю обробки вхідних і вихідних даних, тому вибір оптимальної моделі для конкретної задачі потребує систематичного порівняння на основі відкритих рейтингів, зокрема Chatbot Arena [43]. У контексті задачі порівняння цін LLM відкривають можливість замінити аналіз структури вебсторінок на інтелектуальну інтерпретацію результатів вебпошуку. Мовна модель здатна безпосередньо аналізувати результати пошукової системи та отримувати з них інформацію про ціни. Суттєвою перевагою підходів на основі LLM є відсутність потреби у знанні структури вебсайтів конкретних магазинів та здатність працювати з довільною країною та мовою без зміни програмного коду.

Важливою технічною можливістю сучасних LLM, що суттєво розширює їхнє застосування в автоматизованих системах, є підтримка структурованого виводу. Це означає здатність моделі повертати результат не у формі довільного тексту, а у заздалегідь визначеному форматі, зокрема у форматі JSON відповідно до заданої схеми даних. Структурований вивід дає змогу програмно опрацьовувати результати роботи моделі без додаткового аналізу вільного тексту та безпосередньо інтегрувати їх у подальші етапи автоматизованого конвеєра обробки даних.

Подальшим розвитком застосування LLM є побудова систем на основі агентного підходу [44, 45]. У таких системах мовна модель діє як автономна одиниця, що самостійно планує послідовність дій, викликає зовнішні інструменти (вебпошук, геоінформаційні сервіси, бази даних), спостерігає за результатами та адаптує поведінку залежно від отриманої інформації. На відміну від одноразового виклику мовної моделі із заздалегідь сформованим запитом, агент здатний самостійно визначати, які кроки необхідно виконати для досягнення поставленої мети. Основними обмеженнями підходів на

основі LLM є вартість використання зовнішніх сервісів мовних моделей та залежність від якості результатів пошукових систем.

Незалежно від обраного підходу, вебскрейпінгу чи застосування LLM, для порівняння цін необхідно визначити перелік торговельних мереж, що функціонують у заданому місті. Замість формування та підтримки фіксованих списків магазинів для кожного міста чи країни це завдання може бути автоматизовано за допомогою відкритих картографічних даних. Зокрема проєкт OpenStreetMap [46] та програмний інтерфейс Overpass API [47] дають змогу отримувати перелік магазинів за географічними координатами із зазначенням назви мережі, її вебсайту та кількості локацій.

Таким чином, еволюція підходів до порівняння цін відображає загальну тенденцію переходу від автоматизації збирання даних з вебсторінок до інтелектуальних систем на основі LLM, здатних самостійно інтерпретувати, шукати та перевіряти інформацію.

1.5 Постановка задачі

Таким чином, аналіз сучасного стану справ у галузі моніторингу та порівняння цін на товари свідчить про наявність низки невирішених проблем. Наявні сервіси порівняння цін потребують прямих інтеграцій з кожною торговельною мережею, мають обмежене географічне покриття та не забезпечують можливостей для вбудовування у сторонні інформаційні системи. Жоден з розглянутих сервісів не пропонує механізму порівняння цін на основі скорочених назв товарів із касових чеків, хоча саме касові чеки є одним з найпоширеніших джерел інформації про здійснені покупки. Застосунки для відстеження витрат, що працюють з касовими чеками, використовують розпізнані дані виключно для обліку, але не для пошуку вигідніших цін у інших торговельних мережах. Водночас розвиток LLM та агентного підходу відкриває можливість побудови інтелектуальних систем,

здатних інтерпретувати скорочені назви з касових чеків та здійснювати пошук цін без попередньої інтеграції з конкретними магазинами.

Об'єктом роботи є процес моніторингу та порівняння цін на товари з різних торговельних мереж на основі інформації з касових чеків.

Метою роботи є розроблення інтелектуального модулю для моніторингу цін товарів з паперового чеку та впровадження його у мобільний застосунок для відстеження витрат.

Для досягнення цієї мети необхідно вирішити такі завдання:

- провести аналіз існуючих сервісів порівняння цін та застосунків для відстеження витрат, визначити їхні обмеження;
- проаналізувати підходи до вирішення задачі порівняння цін, зокрема на основі вебскрейпінгу та LLM;
- сформувати набір даних касових чеків для проведення експериментів та оцінки якості роботи системи;
- визначити критерії та метрики оцінки якості пошуку цін;
- розробити та реалізувати прототип підходу на основі вебскрейпінгу з використанням браузерів без графічного інтерфейсу та алгоритмів нечіткого порівняння рядків;
- розробити та реалізувати прототип підходу на основі послідовності LLM із залученням сервісів вебпошуку та завантаження сторінок;
- розробити та реалізувати прототипи агентних підходів за патерном ReAct із використанням спеціалізованих LLM-інструментів;
- провести порівняльний аналіз LLM різних розробників для кожного етапу роботи системи за критеріями якості, швидкості, стабільності та вартості; провести експериментальне порівняння реалізованих підходів та визначити підхід, що найкраще відповідає поставленій задачі;
- сформувати підсумковий алгоритм роботи модулю на основі результатів проведеного порівняння;
- розробити архітектуру інтелектуального модулю для моніторингу цін та впровадити його у мобільний застосунок для відстеження витрат.

2 РОЗРОБКА МЕТОДУ МОНІТОРИНГУ ЦІН НА ТОВАРИ НА ОСНОВІ ІНФОРМАЦІЇ З КАСОВИХ ЧЕКІВ

2.1 Постановка завдання та критерії оцінки якості

Метою цього розділу є експериментальне порівняння чотирьох підходів до автоматизованого пошуку цін на товари з касових чеків та обґрунтування вибору одного з них для подальшої програмної реалізації. Порівняння виконують на єдиному власному наборі даних. На вхід кожному методу подають лише назву позиції з чека, місто та код країни. Еталонні дані з чека (фактичний магазин і ціну покупки) методам не передають. Результати кожного підходу оцінюють вручну. Кожному знайденому запису присвоюють ознаку відповідності товару (точна відповідність, близька відповідність, інший варіант того самого товарного ряду або нерелевантний запис) та ознаку достовірності ціни. Запис вважається релевантним, якщо він є точною або близькою відповідністю шуканому товару з підтвердженою ціною.

Підходи порівнюють за трьома осями: якість пошуку, часові витрати та вартість обробки.

Якість пошуку оцінюють за такими показниками:

- точність (англ. precision) – частка релевантних записів серед усіх повернутих методом;
- строга точність – частка лише точних відповідностей з підтвердженою ціною серед усіх повернутих записів;
- охоплення (англ. coverage) – середня кількість унікальних релевантних магазинів на один товар;
- правильна найдешевша ціна – частка товарів, для яких найдешевший повернутий запис є точною відповідністю з підтвердженою ціною, що безпосередньо відображає основне призначення модуля допомогти користувачеві обрати вигіднішу пропозицію;

– чесність самооцінки – серед записів, які метод сам позначив як точну відповідність, частка справді точних (лише для підходів на основі LLM).

Часові витрати – середній час повного циклу обробки однієї позиції чека, обчислений за всіма позиціями набору даних.

Вартість обробки – середні грошові витрати у доларах США на обробку однієї позиції чека з урахуванням оплати викликів LLM, вебпошуку та завантаження сторінок. Для підходу на основі вебскрейпінгу прямі грошові витрати на зовнішні сервіси відсутні.

2.2 Створення власного набору даних чеків з ідентичними товарами з різних торговельних мереж

Готових публічних наборів даних, що містили б скорочені назви товарів з касових чеків з підтвердженими цінами в різних торговельних мережах, на момент виконання роботи знайти не вдалось, тому збір даних проведено самостійно на основі реальних касових чеків з мереж Словаччини, Австрії, Молдови та України.

Набір даних має ієрархічну структуру і складається з трьох частин, кожна з яких призначена для окремого типу експериментів.

Перша частина – наскрізний набір з 20 позицій, призначений для тестування методу в цілому. Товари відібрано за принципом максимального покриття типових сценаріїв обробки скорочених назв: різні країни, мови, категорії товарів, типи аббревіатур та структури запису. Набір збалансовано за країнами (7 позицій зі Словаччини, 7 – з України, 6 – з Австрії) та за наявністю бренду. Представлено міжнародні бренди (Bounty, Coca-Cola, Lindt, Twinings, Bonduelle), локальні бренди (Rajo, Pilos, Садочок, Кулиничі, Stauds) і товари без вказаного бренду (Lkft. mlieko, Jablko Evelina, Гриби Печериці, Томати коктейльні). Повний перелік позицій наведено у Додатку А (табл. А.1). Цей набір використовується для повного порівняння

чотирьох підходів, де кожна позиція проходить через них усі з обчисленням метрик якості, часу та вартості.

Друга частина – набір з 40 позицій для вибору моделей окремих компонентів послідовності LLM. Сформовано як розширення першого зі збільшеною різноманітністю типів аббревіатур, мов та країн. До набору входять по 10 позицій з касових чеків кожної з чотирьох країн (Словаччини, Австрії, Молдови та України). Включення австрійського та молдовського сегментів додає до слов'янської мовної групи покриття германської (німецької) та романської (молдовської / румунської) груп. Молдовська частина використовується переважно для локального тестування етапу нормалізації – повний цикл обробки для неї обмежено через менш розвинений онлайн-сегмент роздрібної торгівлі в Молдові. Приклади позицій з кожної країни наведено у таблиці 2.1

Таблиця 2.1 – Приклади позицій набору даних за країнами

Країна	Приклади позицій
Словаччина	Lktf. mlieko; Mlieko Rajo Bezl 1,5 % 1l; Jablko Evelina; Bounty 57g; Losos 4×125g; Pilos Tvar. 0,5 %; Par. San Marzano
Австрія	LINDT ALKOHOL EIER SORTIE; Twin. Engl.Breakf.; Bonduelle Bio Mais; COKE ZERO ZERO 0,33L; MANDARINE SPAR GELEG; STAUDS LIMITIERTE HIMBEER
Молдова	Iaurt Dolce piersic/maracuja 115g; Smantana President 15 % 350g pah.; Nectar Vita piersici/mango 200ml; Mere rosii kg Moldova; Bautura carb. Coca-Cola 1.75L pet; Branzica Florica glaz.coac.neagra/zm.20% 36g
Україна	Напій 1 л Sprite ПЕТ; Батон 0,5 кг Кулиничі Дорожній; Гриби Печериці фас 0,5 кг 1 гат; Томати коктейльні 1 гат; Соковий напій 0,95 л Садочок Мультифрукт; Морозиво 0,6 кг Ласунка «Mix Deluxe»; Сіль 1 кг Вибір кам'яна мелена

Третя частина – набір з 60 розмічених записів для вибору моделі етапу перевірки. Кожен запис є парою зі скороченої назви товару з касового чека (наприклад, «Mlieko Rajo Bezl 1,5 % 1l») та одного з варіантів-кандидатів,

знайдених в онлайн-каталозі торговельної мережі (наприклад, «Mlieko Ražo Bezlaktózové polotučné 1,5 % UHT 1 l»).

Для побудови набору обрано 6 позицій з першої частини. Для кожної зібрано по 10 кандидатів так, щоб охопити три типові випадки: точне співпадіння з шуканим товаром, частковий збіг (інша жирність, об'єм, добавка) та помилкове співпадіння (інший товар тієї самої категорії). Кожну з 60 пар вручну розмічено за чотирма ознаками (бренд, підтип, об'єм, додаткові атрибути) з підсумковою позначкою про відповідність пари за всіма ознаками. Така схема дозволяє оцінити не лише підсумкову якість моделей на цьому етапі, але й визначити, на якій ознаці модель найчастіше припускається помилки.

Фотографії касових чеків з характерними позиціями наведено у Додатку А: словацьке безлактозне молоко з мереж Lidl та COOP Jednota, Братислава (рис. А.1, А.2); шоколадний набір «Alcohol Eggs Mix» з трьох мереж у Словаччині та Австрії (рис. А.3 – А.5); український соковий напій «Садочок Мультифрукт 0,95 л» з мережі АТБ, Харків (рис. А.6).

2.3 Підхід на основі вебскрейпінгу з браузером без графічного інтерфейсу

Перший підхід реалізовано на основі класичного вебскрейпінгу. Цей підхід не використовує LLM і базується на спрощених правилах обробки тексту: автоматичне завантаження вебсторінок магазинів через браузер без графічного інтерфейсу (англ. headless-браузер), виокремлення інформації про товари та ціни з вмісту сторінок, нечітке зіставлення знайдених товарів з шуканим за допомогою алгоритмів порівняння рядків. Підхід відображає ранній етап розвитку галузі, коли мовні моделі для роботи з природною мовою ще не були широко доступні; він обраний як базовий, відносно якого порівнюються сучасніші підходи.

2.3.1 Архітектура та компоненти підходу

Робота підходу складається з чотирьох послідовних етапів, кожен з яких реалізовано окремим функціональним компонентом. Узагальнену схему підходу наведено у Додатку Б (рис. Б.1). На першому формується перелік торговельних точок продовольчого профілю у місті покупки через картографічну базу OpenStreetMap [46] із запитом за координатами та категоріями. На другому для кожного магазину виконується спроба знайти товар на його сайті через групу браузерів без графічного інтерфейсу з модифікованою бібліотекою автоматизації [37, 38] та механізмом обходу Cloudflare. Реалізовано два методи, а саме прямий пошук на сайті магазину та пошук через зовнішню пошукову систему з обмеженням за доменом. На третьому для кожної знайденої назви обчислюється показник схожості з назвою з чека за метрикою нечіткого порівняння рядків [40, 41] з ваговими внесками за спільним брендом та числовими атрибутами; кандидати з показником нижче порога відкидаються. На четвертому результати позбавляються повторів, з кожного магазину обирається найкращий кандидат та формується підсумковий перелік.

2.3.2 Експериментальна перевірка

Підхід на основі вебскрейпінгу випробувано на всіх 20 позиціях набору даних (підрозділ 2.2, таблиця А.1). Перелік магазинів для кожної позиції отримано з картографічної бази OpenStreetMap. Обхід виконували лише по магазинах з найповнішим описом. Для Братислави та Харкова це до 25 магазинів, для Хайнбургу-на-Дунаї до 8, оскільки розширення переліку пропорційно збільшує й без того значний час обробки. Менша кількість магазинів для Австрії додатково зумовлена примусовим звуженням пошуку до національної доменної зони «.at» у радіусі 7 км. Без такого обмеження

картографічна служба у прикордонному місті повертала переважно магазини сусідніх країн.

Релевантність кожного знайденого запису визначено вручну за критеріями підрозділу 2.1. Точність обчислено як частку релевантних записів серед усіх знайдених підходом. Детальні результати за окремими позиціями наведено у Додатку В (табл. В.1).

Якісний аналіз результатів за окремими позиціями виявив три типові класи поведінки підходу.

Випадок успішного зіставлення – позиції «Par. San Marzano», «Гриби Печериці фас 0,5 кг 1 гат» та «Twin. Engl.Breakf.» (точність 100 %). Усі знайдені на них записи підтверджено як релевантні. Збіг за брендом, типом товару та числовими атрибутами забезпечив високий показник схожості без хибних спрацювань.

Випадок хибного зіставлення – позиції «Lktf. mlieko», «Pilos Tvar. 0,5 %» та «MANDARINE SPAR GELEG». На «Lktf. mlieko» (безлактозне молоко без зазначеного бренду) жоден з 83 кандидатів не пройшов поріг схожості: коротка узагальнена назва без бренду й чисел не дає алгоритму нечіткого порівняння достатнього показника. На «Pilos Tvar. 0,5 %» серед знайдених записів опинилася сторінка косметичного товару (валика для догляду за шкірою), хибно зарахована за збігом числа «0,5» та фрагмента назви. Це характерний приклад нездатності алгоритму розрізняти товари за змістом, а не за поверхневою схожістю рядків.

Випадок часткового зіставлення – позиції «Соковий напій 0,95 л Садочок Мультифрукт» (33 %), «Jablko Evelina» (25 %) та «Losos 4×125g» (25 %). Серед знайдених записів переважають варіанти того самого товарного ряду (інший смак, об'єм чи фасування) та категорійні сторінки з недостовірною ціною. Зокрема, для «Losos 4×125g» одну з категорійних сторінок алгоритм зарахував із ціною 100,0, що не відповідає фактичній вартості товару, тож таку ціну визнано недостовірною.

Окремо слід відзначити вплив жорсткого порога фільтрування на правильні записи. Для позиції «Батон 0,5 кг Кулиничі Дорожній» правильний запис у мережі «Сільпо» отримав показник схожості 34 одиниці й не подолав поріг 40, через що підхід повернув лише один магазин замість кількох доступних. Це демонструє системний компроміс підходу, де підвищення порога зменшує кількість хибних спрацювань, але водночас відсікає й частину правильних записів.

2.3.3 Висновки щодо точності, часових витрат та вартості

Точність. На всіх 20 позиціях підхід повернув 62 записи із середньою точністю 50 % (частка релевантних серед усіх повернутих) та строгою точністю 39 % (лише точні відповідності з підтвердженою ціною). Охоплення становить 1,6 релевантного магазину на товар. Правильну найдешевшу ціну підхід повернув для 47 % товарів. Для однієї позиції не знайдено жодного запису. Структура помилок має системний характер. Алгоритм нечіткого порівняння зіставляє рядки за поверхневою схожістю, а не за змістом, тому зараховує сторонні товари зі спільними словами чи числами та не відсіює категорійні сторінки з недостовірною ціною. Додатковим джерелом втрат є жорсткий поріг фільтрування, який разом із хибними спрацюваннями відсікає й частину правильних записів з низьким показником схожості.

Часові витрати. Середній час обробки однієї позиції становить 369 секунд, сумарний час прогону по 20 позиціях становить близько 123 хвилин. Час лінійно залежить від кількості магазинів, для яких виконується пошук, і зумовлений одночасним утриманням кількох екземплярів повноцінного браузера та очікуванням завантаження динамічних сторінок. Його зменшення до 1–2 хвилин на позицію можливе за рахунок зменшення кількості магазинів, що пропорційно зменшить охоплення.

Вартість. Прямі грошові витрати на зовнішні сервіси відсутні, оскільки використовує відкриті картографічні дані та власні браузері без графічного інтерфейсу й не звертається до платних програмних інтерфейсів. Натомість він має нетривіальні вимоги до обчислювальних ресурсів через паралельну роботу кількох браузерів.

Стійкість до механізмів виявлення автоматизованого доступу. За весь прогін підхід жодного разу не був заблокований захисним екраном Cloudflare; знайдені ціни походять із реальних сторінок торговельних мереж (Billa, Spar, Сільпо, АТБ та інших), що підтверджує технічну здатність браузерного підходу обходити сучасні системи виявлення.

Підхід на основі вебскрейпінгу не потребує прямих грошових витрат і здатний обходити системи виявлення автоматизованого доступу, проте має низьку точність, вузьке охоплення та значний час обробки. Його принциповим обмеженням є нездатність до семантичної інтерпретації скорочень і варіантів товару, що для задачі моніторингу цін з касового чека є визначальним. З огляду на це підхід розглядається як базовий, відносно якого оцінюються підходи на основі LLM.

2.4 Підхід на основі чіткої послідовності великих мовних моделей

Другий підхід реалізовано на основі LLM з вбудованим інструментом вебпошуку. На відміну від вебскрейпінгу, цей підхід не вимагає попереднього знання структури вебсайтів конкретних магазинів та здатний працювати з довільною країною та мовою без зміни програмного коду. У межах цього підходу процес моніторингу цін представлено у вигляді чіткої послідовності LLM, кожна з яких виконує окрему спеціалізовану задачу обробки тексту та видобування інформації.

2.4.1 Архітектура послідовності та її компоненти

Процес моніторингу цін представлено у вигляді чіткої послідовності з шести етапів, визначеної заздалегідь і незалежної від результатів проміжних обчислень. Кожен етап має визначений програмний інтерфейс, що дає змогу замінити модель на окремому етапі без впливу на сусідні та забезпечує незалежне оцінювання якості кожного компонента.

Узагальнену схему послідовності наведено у Додатку Б (рис. Б.2).

Послідовність складається з шести етапів:

- нормалізація запиту – перетворення скороченого рядка з чека на структурований запис з виокремленими полями (бренд, тип, підтип, обсяг, атрибути) та генерування пошукових запитів для подальших етапів;
- широкий вебпошук – збирання сирого обсягу текстових результатів вебпошуку за згенерованими на попередньому етапі запитам з прив'язкою фрагментів до URL-джерел;
- виокремлення цінових даних – перетворення сирого тексту результатів пошуку на структурований перелік знахідок (магазин, товар, ціна, валюта, посилання);
- цільове доповнення – додаткові пошукові запити до магазинів, не представлених у зібраних результатах, з використанням переліку торговельних мереж міста з відкритих картографічних даних OpenStreetMap;
- уточнення знайдених цін на повних сторінках джерел – завантаження повної сторінки за кожним посиланням через зовнішній сервіс видобування вмісту та повторне видобування актуальної ціни безпосередньо зі сторінки;
- перевірка знайдених записів – оцінювання кожної знахідки за чотирма критеріями (бренд, підтип, обсяг, атрибути) та обчислення підсумкової оцінки довіри з розмежуванням знахідок на категорії точної відповідності, схожої знахідки та низької довіри.

2.4.2 Принципи відбору моделей для експериментів

Чітка послідовність дає змогу для кожного етапу окремо обрати оптимальну модель. Проведено експерименти з вибору моделі для трьох етапів різних класів задач, а саме нормалізації запиту, широкого вебпошуку та перевірки знайдених записів. Решта етапів за характером задачі належать до того ж класу, що й нормалізація, тому використовують ту саму модель. Для первинного відбору моделей-кандидатів використано рейтинг Arena AI [43] із двома категоріями: Search (для етапу вебпошуку) та Instruction Following (для текстових етапів). У категорії Instruction Following моделі додатково поділено на три цінових рівні від кожного постачальника (порогові ціни наведено у Додатку Г). Це дає змогу виявити точку насичення якості відносно ціни. Скріншоти інтерфейсу з налаштованими фільтрами наведено у Додатку Г (рис. Г.1–Г.3). Оцінювання моделей здійснюється за трьома осями: якість, час одного виклику та вартість обробки. Конкретні моделі, відібрані для кожного з трьох етапів, наведено у Додатку Д.

2.4.3 Зміст компонентів послідовності

Кожен етап реалізовано окремою LLM зі спеціалізованим системним промптом та структурованою JSON-схемою вихідних даних [48]. Опис шести етапів з повною специфікацією вхідних і вихідних даних та особливостей реалізації наведено у Додатку Е.

2.4.4 Експерименти з вибору моделей

Для трьох ключових етапів послідовності проведено експериментальне порівняння великих мовних моделей різних постачальників. Порівняно

моделі для етапу нормалізації запиту, етапу широкого вебпошуку та етапу перевірки знайдених записів (опис етапів наведено у Додатку Е). Вибір саме цих трьох етапів зумовлений їх ключовою роллю у роботі послідовності. Помилка на нормалізації веде до пошуку зовсім іншого товару. Неякісний вебпошук залишає більшість магазинів поза знахідкою. Помилки на перевірці визначають, які знахідки потраплять до користувача як «точні відповідності».

Для кожного з трьох етапів використано специфічні метрики якості: коефіцієнти Дайса та Жаккара для нормалізації, точність, охоплення та $F1$ -міра для вебпошуку, точність класифікації за чотирма критеріями для перевірки знайдених записів.

Для категорії Instruction Following застосовано поділ моделей на три цінові рівні (дешевий, середній, преміальний) з правилом відбору топ-3 моделі рейтингу Arena AI Instruction Following у кожному рівні з покриттям усіх великих постачальників API. Для категорії Search поділ за цінами не застосовано через переважання моделей преміального рівня в цій категорії; обрано топ-10 моделей рейтингу Arena AI Search з покриттям усіх великих постачальників. Сформовані переліки моделей-кандидатів для категорій Instruction Following та Search наведено у Додатку Ж (табл. Ж.1, Ж.2).

Опис методики порівняння, метрик оцінювання, фільтрів придатності, повні результати з аналізом помилок моделей для кожного з трьох етапів та підсумковий набір обраних моделей для всіх шести етапів наведено у Додатку Д (табл. Д.1–Д.4).

2.4.5 Експериментальна перевірка послідовності

Послідовність LLM випробувано на всіх 20 позиціях набору даних (табл. А.1) з моделями, обраними за результатами порівняння (підпункт 2.4.4, Додаток Д). Релевантність кожного знайденого запису

визначено вручну за критеріями підрозділу 2.1. Детальні результати за окремими позиціями наведено у Додатку В (табл. В.2).

На відміну від вебскрейпінгу, послідовність працює за принципом максимального охоплення: вона завантажує значну кількість сторінок-джерел (від 6 до 35 на позицію) і після виокремлення та валідації цін формує підсумковий перелік (від 5 до 25 записів на позицію). Для жодної позиції підхід не залишився без знахідок. Якісний аналіз результатів виявив три типові класи поведінки.

Широке охоплення з прийнятною точністю – позиції з чітким описом або відомим брендом («Напій 1 л Sprite ПЕТ» – 68 %, «COKE ZERO ZERO 0,33L» – 68 %, «STAUDS LIMITIERTE HIMBEER» – 73 %, «Lkft. mlieko» – 80 %). На таких товарах підхід знаходить ціни в багатьох мережах, і більшість знахідок є релевантними.

Широке охоплення з низькою точністю – позиції, на яких у видачу потрапляють варіанти того самого ряду та сторонні джерела («Pilos Tvar. 0,5 %» – 27 %, «Bonduelle Bio Mais» – 38 %, «Соковий напій 0,95 л Садочок Мультифрукт» – 42 %, «MANDARINE SPAR GELEG» – 0 %). Серед знахідок трапляються інші варіанти лінійки (інша жирність, смак чи об'єм) та нерелевантні джерела на кшталт маркетплейсів і закладів харчування, які модель помилково зараховує як цінові пропозиції.

Помилкова найдешевша ціна – окремий клас, критичний для задачі моніторингу. На низці позицій найдешевшою серед знайдених виявляється нерелевантний запис. Для «Батон 0,5 кг Кулиничі Дорожній» це оголошення стороннього продавця, для «Гриби Печериці фас 0,5 кг 1 гат» сторінка закладу харчування, для «Томати коктейльні 1 гат» лот маркетплейсу. Через це підхід пропонує користувачеві хибну найнижчу ціну, що особливо помітно на українських сайтах.

2.4.6 Висновки щодо точності, часових витрат та вартості

Підсумковий набір моделей для всіх шести етапів послідовності за результатами порівняння наведено у Додатку Д (табл. Д.4).

Якість послідовності перевірено на двох рівнях. На рівні окремих етапів (Додаток Д) обрані моделі забезпечують AVG_w нормалізації 0,787. Ця метрика обчислюється як зважене середнє шести метрик подібності полів нормалізованого опису товару. Для текстових полів (бренд, тип товару, тип англійською, підтип) застосовується коефіцієнт Дайса, для множинних полів (обсяг або вага, додаткові атрибути) – коефіцієнт Жаккара. Поля бренда та обсягу мають подвоєну вагу як найкритичніші для ідентифікації товару. F1-міру вебпошуку 0,444 та загальну точність перевірки знайдених записів 80,1 % з часткою пропусків 1,7 %. На наскрізному рівні (підпункт 2.4.5) повна послідовність на 20 позиціях повернула 262 записи із середньою точністю 57 % та строгою точністю 39 %. Охоплення становить 7,1 релевантного магазину на товар. Правильну найдешевшу ціну підхід повернув для 55 % товарів; для жодної позиції не залишився без знахідок. Чесність самооцінки становить 53 %. Це означає, що серед записів, які послідовність позначила як точну відповідність, лише трохи більше половини справді є такими за ручною розміткою.

Часові витрати. Середній час обробки однієї позиції становить 119 секунд, сумарний час прогону по 20 позиціях – близько 40 хвилин.

Вартість. Середня вартість обробки однієї позиції становить 0,146 долара, сумарна вартість прогону по 20 позиціях – 2,92 долара.

Порівняно з базовим підходом на основі вебскрейпінгу (підрозділ 2.3) послідовність LLM забезпечує суттєво ширше охоплення магазинів (7,1 проти 1,6 релевантного магазину на товар), вищу точність (57 % проти 50 %) та менший час обробки (119 проти 369 секунд на позицію). Платою за ці переваги є експлуатаційна вартість близько 0,146 долара на позицію, тоді як вебскрейпінг не потребує грошових витрат на зовнішні сервіси.

Серед сильних сторін підходу:

- універсальність по країнах та мовах і стійкість до зміни структури вебсайтів завдяки семантичній обробці замість прив'язки до мови гіпертекстової розмітки (англ. HyperText Markup Language, HTML);
- модульність архітектури – заміна моделі на окремому етапі без впливу на решту (обґрунтування вибору моделей наведено у підпункті 2.4.4 та Додатку Д);
- широке охоплення магазинів та відсутність позицій без знахідок;
- швидкий час обробки (119 секунд на позицію).

Серед обмежень:

- низька чесність самооцінки (53 %) – система часто позначає як точну відповідність записи, що насправді є близькими або нерелевантними;
- великий розрив між точністю та строгою точністю (57 % проти 39 %) через значну частку варіантів у видачі;
- правильна найдешевша ціна лише для 55 % товарів через потрапляння нерелевантних джерел (маркетплейсів, закладів харчування) у видачу;
- вартість 0,146 долара на позицію формує постійні експлуатаційні витрати пропорційно навантаженню;
- залежність від зовнішніх постачальників API (OpenAI, Google, Anthropic, Firecrawl) – їхньої доступності, тарифних змін та оновлення моделей;
- поширення похибок між етапами – помилка нормалізації або вебпошуку впливає на всі подальші етапи без механізму зворотного зв'язку.

2.5 Підхід на основі інтелектуального агента

Третій підхід реалізовано на основі інтелектуального агента, тобто LLM, який сам обирає послідовність кроків залежно від проміжних

результатів та має доступ до набору зовнішніх інструментів. На відміну від чіткої послідовності з підрозділу 2.4, де порядок етапів та інструментів визначений заздалегідь, агент визначає кожен наступний крок на основі попередніх результатів.

2.5.1 Концепція агентного підходу

В основі підходу лежить цикл міркування, дія, спостереження (англ. reasoning, action, observation), відомий у літературі як ReAct (акронім від англ. Reasoning та Acting) [44]. На кожній ітерації агент аналізує накопичену інформацію, формулює міркування щодо наступного кроку, обирає один з доступних інструментів та подає до нього аргументи. Виклик інструменту повертає результат, який додається до контексту наступної ітерації [52]. Цикл триває до виклику фінального інструменту завершення або до досягнення граничної кількості ітерацій.

Адаптивна поведінка дозволяє агенту враховувати специфіку конкретного товару та контексту. Наприклад, для нелокального бренду агент може спочатку перевірити присутність товару на агрегаторах, а для локальної марки одразу звернутися до сайтів місцевих мереж. Такий вибір не визначений у програмному коді. Він є результатом міркування моделі на основі вже зібраних спостережень.

2.5.2 Архітектура агента

Агент побудований навколо однієї LLM, що виконує всі кроки міркування та вибору інструментів, та чотирьох зовнішніх інструментів, описаних у наступному підпункті. Узагальнену схему агента наведено у Додатку Б (рис. Б.3). На вхід агент отримує оригінальний рядок з касового

чека, код країни покупки та назву міста. На виході він повертає структурований перелік знахідок цін, аналогічний до результатів підходу на основі послідовності.

Конфігураційні параметри агента:

- модель – gpt-5-mini як така, що поєднує достатню якість для задач планування з помірною вартістю одного виклику;
- температура генерації – 0, для максимальної відтворюваності рішень агента;
- гранична кількість ітерацій – 30, що обмежує сумарний час обробки одного запиту та вартість у випадках, коли агент не здатний зійтись до фінальної відповіді;
- обов’язкове поле з описом міркувань у структурованій відповіді на кожному кроці, що забезпечує прозорість міркувань і дає змогу ретроспективно аналізувати рішення агента.

Системний промпт містить опис задачі, перелік доступних інструментів з їх сигнатурами, рекомендовану стратегію (спочатку отримати перелік місцевих магазинів, потім виконувати пошук, при потребі завантажувати окремі сторінки) та правила завершення (викликати завершальний інструмент тільки коли зібрано достатньо релевантних знахідок).

2.5.3 Доступні інструменти агента

Агенту надано чотири інструменти, кожен з яких виконує одну конкретну дію.

Інструмент `list_local_stores(city, country_code)` повертає перелік торговельних мереж у заданому місті, отриманий з відкритих картографічних даних OpenStreetMap через сервіс запитів Overpass API. Перелік ранжується

за кількістю фізичних точок мережі. Інструмент використовується агентом на початку для отримання уявлення про доступні джерела цін у регіоні.

Інструмент `search(query, country_code)` виконує вебпошук через зовнішній сервіс Tavily [53] і повертає перелік результатів з назвою сторінки, фрагментом тексту та URL. На відміну від підходу 2.4, агент отримує сирий перелік фрагментів і сам їх інтерпретує.

Інструмент `fetch_page(url)` завантажує повну сторінку за вказаним URL через сервіс Firecrawl [54] і повертає її вміст у вигляді markdown-розмітки. Сервіс здатний відтворювати сторінки з динамічним вмістом, обходити захисні механізми вебсайтів, зокрема Cloudflare, та коректно обробляти односторінкові вебзастосунки (англ. Single Page Application, SPA).

Інструмент `submit_final_answer(matches, summary)` є завершальним: агент викликає його, коли вважає, що зібрано достатньо знахідок. Аргумент `matches` містить структурований перелік знайдених цін з ознаками відповідності, а аргумент `summary` містить короткий узагальнюючий опис результату для користувача.

2.5.4 Експерименти з вибору моделі агента

У підході на основі простого агента одна LLM виконує всю роботу: координує виклики інструментів, аналізує результати та формує підсумкову оцінку знайдених варіантів. Тому вибір цієї моделі потребує окремого експериментального обґрунтування.

Для звуження переліку кандидатів використано рейтинг Arena AI у категорії Instruction Following [43] з ціновими фільтрами Input до 2,50 та Output до 15 доларів за мільйон токенів (топ цієї категорії від OpenAI та Google наведено у Додатку Г, рисунки Г.5 і Г.6). Замість відбору фіксованої кількості найкращих моделей за рейтингом застосовано вибірку трьох цінових рівнів від кожного постачальника (дешевий, середній, преміальний),

що дає змогу виявити точку насичення якості відносно ціни. За підсумками відбору протестовано шість моделей: від OpenAI – gpt-5-mini, gpt-5.1, gpt-5.2; від Google – gemini-3.1-flash-lite, gemini-3-flash-preview, gemini-2.5-pro.

Кожну модель-кандидат тестовано на наборі з шести товарів (трьох словацьких та трьох українських) при температурі генерації 0 та максимальній кількості ітерацій 20. Усього виконано 36 запусків (6 моделей по 6 товарів). Правильність кожної знайденої агентом ціни визначено через ручну розмітку. Повні результати порівняння наведено у Додатку И (табл. И.1).

За результатами обрано модель gpt-5-mini: точність 0,69, чесність самооцінки 0,82, правильна найдешевша ціна у 83 % запусків, середня кількість правильно знайдених цін на запуск 5,5, середня вартість одного запуску 0,130 долара, середній час 191 секунда. Лише модель gpt-5.1 перевершує gpt-5-mini за точністю (0,75) та чесністю (0,87), однак повертає правильну найдешевшу ціну лише у половині запусків і має на 72 % вищу вартість (0,224 долара). Моделі Google поступаються gpt-5-mini за точністю (0,60–0,65 проти 0,69). Найнижчий показник правильної найдешевшої ціни зафіксовано у gemini-3-flash-preview, лише 17 % товарів.

2.5.5 Приклад роботи агента на типовому запиті

Для ілюстрації роботи інтелектуального агента у Додатку К наведено повну послідовність кроків на представницькому запиті «Mlieko Ražo Bezl 1,5 % 1l» у місті Братислава. Запит обрано як характерний для предметної області, а саме словацький локальний бренд з аббревіатурою «Bezl», що потребує розкриття у «bezlaktózové».

Наведена послідовність демонструє характерну поведінку агента. Агент чергує загальні пошуки з цільовими запитами до конкретних доменів, виконує повторні спроби з різними формулюваннями для тих самих мереж та

вибірково завантажує сторінки лише за потреби. Жоден з цих кроків не визначений у коді явно. Усі рішення прийняті LLM на основі попередніх спостережень.

2.5.6 Експериментальна перевірка агента

Простий агент випробувано на всіх 20 позиціях набору даних (табл. А.1). Релевантність кожного знайденого запису визначено вручну за критеріями підрозділу 2.1. Детальні результати за окремими позиціями наведено у Додатку В (табл В.3). Для ілюстрації роботи агента у Додатку К наведено повну послідовність кроків на типовому запиті «Mlieko Rajo Bezl 1,5 % 1l» у місті Братислава.

Характерно, що для більшості позицій агент повертає рівно 8 записів, після чого завершує пошук. Для п'яти позицій він не знайшов жодного підтвердженого запису за 30 дозволених кроків. Якісний аналіз результатів виявив три типові класи поведінки.

Високоточний пошук спостерігається для позицій з відомим брендом або чіткою специфікацією, де агент знаходить записи із точністю 88–100 %. Серед них «Mlieko Rajo Bezl 1,5 % 1l» (100 %), «Bounty 57g» (100 %), «Lkth. mlieko» (100 %), «Par. San Marzano» (100 %), «Twin. Engl. Breakf.» (100 %).

Вичерпання ліміту ітерацій спостерігається п'ять позицій, на яких агент використав усі 30 дозволених кроків і не повернув жодного запису. Це «Losos 4×125g», «Pilos Tvar. 0,5 %», «LINDT ALKOHOL EIER SORTIE», «Bonduelle Bio Mais» та «MANDARINE SPAR GELEG». Спільна риса цих товарів – короткий або неоднозначний запис на чеку, наприклад, аббревіатура приватної марки (Pilos), скорочена назва (MANDARINE SPAR GELEG) або вузький асортимент в онлайн-магазинах регіону. Агент витрачає ітерації на безрезультатні пошуки та завантаження сторінок, які не дають підтвердженого збігу.

Правильні записи з помилковою найдешевшою ціною – позиції, де більшість знахідок релевантні (75–88 %), але найдешевший запис виявляється іншим варіантом або має непідтверджену ціну. Наприклад, для «Батон 0,5 кг Кулиничі Дорожній» найдешевшим є інший варіант, для «Гриби Печериці фас 0,5 кг 1 гат» найдешевшим є близька, але не точна відповідність, а для «COKE ZERO ZERO 0,33L» найдешевшою виявилась звичайна Coca-Cola замість Zero. Навіть при високій загальній точності агент не завжди розрізняє найтонші варіантні відмінності серед знахідок.

Загалом агент повертає меншу кількість записів порівняно з послідовністю LLM (підрозділ 2.4), проте з суттєво вищою часткою релевантних серед них. Послідовний характер ReAct-циклу, де кожне рішення про наступну дію приймається після завершення попередньої, пояснює значний час обробки (у середньому 335 секунд на позицію) та високу варіативність часу між товарами (від 181 до 592 секунд).

2.5.7 Висновки щодо точності, часових витрат та вартості

Точність. На всіх 20 позиціях простий агент повернув 121 запис із середньою точністю 88 % та строгою точністю 71 %. Охоплення становить 5,3 релевантного магазину на товар. Правильну найдешевшу ціну підхід повернув для 67 % товарів (серед тих, для яких знайдено хоча б один запис). Чесність самооцінки становить 88 %, що значно вище, ніж у послідовності LLM (53 %), і свідчить про точнішу внутрішню оцінку збігів. Водночас п'ять позицій залишилися без жодної знахідки, оскільки агент вичерпував ліміт ітерацій, не знайшовши підтвердженого збігу.

Часові витрати. Середній час обробки однієї позиції становить 335 секунд, сумарний час прогону по 20 позиціях – близько 112 хвилин.

Вартість. Середня вартість обробки однієї позиції становить 0,185 долара, сумарна вартість прогону по 20 позиціях – 3,69 долара.

Порівняно з послідовністю LLM (підрозділ 2.4) простий агент забезпечує суттєво вищу точність (88 % проти 57 %) та строгу точність (71 % проти 39 %), а також вищу правильну найдешевшу ціну (67 % проти 55 %). Натомість він повільніший (335 проти 119 секунд) і дорожчий (0,185 проти 0,146 долара на позицію). Охоплення становить 5,3 магазину на товар, що менше, ніж у послідовності (7,1), але суттєво більше, ніж у вебскрейпінгу (1,6, підрозділ 2.3). Головною платою за вищу точність є п'ять позицій без знахідок (проти нуля у послідовності).

Серед сильних сторін підходу:

- суттєво вища точність і строга точність порівняно з послідовністю та вебскрейпінгом;
- адаптивна стратегія пошуку, де агент автономно обирає інструменти та формує запити на підставі проміжних результатів;
- здатність до самокорекції, тобто повторний пошук або зміна стратегії у разі невдачі попередньої спроби;
- висока чесність самооцінки (88 %);
- прозорість міркувань, де обов'язковий опис міркувань на кожному кроці забезпечує можливість ретроспективного аналізу рішень агента.

Серед обмежень:

- п'ять позицій без знахідок (25 % набору) через вичерпання ліміту ітерацій на складних або неоднозначних товарах;
- значний час обробки (335 секунд на позицію) через послідовний характер ReAct-циклу;
- висока варіативність часу між товарами (від 181 до 592 секунд), що ускладнює прогнозування часу відповіді у виробничому використанні;
- вартість 0,185 долара на позицію, що вища за послідовність LLM (0,146 долара);
- залежність якості від можливостей єдиної моделі-оркестратора, оскільки усі етапи (пошук, аналіз сторінок, верифікація) покладено на одну модель без спеціалізації.

2.6 Підхід на основі гібридного агента над послідовністю LLM

2.6.1 Концепція гібридного підходу

Підхід на основі чіткої послідовності LLM (підрозділ 2.4) демонструє високу якість на окремих етапах за рахунок фіксованої послідовності спеціалізованих моделей, але не передбачає прийняття рішень про послідовність викликів, оскільки кожен етап обробляє всі вхідні дані, які отримує. Підхід на основі простого агента (підрозділ 2.5) додає гнучкість прийняття рішень за рахунок агента-оркестратора, проте делегує всю обробку даних безпосередньо моделі-оркестратору через інженерні інструменти, що збільшує навантаження на одну модель та обсяг переданих їй даних.

Гібридний підхід об'єднує сильні сторони обох рішень. Агент-оркестратор приймає рішення про послідовність дій, як у підході на основі простого агента. При цьому специфічні задачі обробки тексту (нормалізація запиту, виокремлення цін, перевірка ціни на сторінці, оцінка впевненості) винесено в окремі інструменти, кожен з яких є самостійним викликом LLM зі спеціалізованою моделлю, як у підході на основі послідовності. Це дозволяє кожному етапу обробки використовувати найкращу для нього модель, водночас зберігаючи можливість агента-оркестратора відхилитися від типового маршруту за наявності аномалій, наприклад, перевірити підозрілу ціну на сторінці перед включенням її у фінальний результат.

2.6.2 Інструменти агента

Агенту надано сім інструментів, розділених на дві категорії за способом виконання.

LLM-інструменти, що відтворюють окремі етапи послідовності з підрозділу 2.4:

– `normalize_product(item_name, source_store, ...)` – нормалізація запиту з касового чека у структурований формат; модель `gpt-4.1-mini-2025-04-14` (за результатами Додатку Д);

– `extract_prices(block_ids, normalized_query)` – виокремлення пар «магазин – ціна» з фрагментів результатів пошуку та узагальнення цін у агрегаторах; модель `gpt-4.1-mini-2025-04-14`;

– `verify_price(url, expected_price)` – повторна перевірка ціни на сторінці магазину через її завантаження сервісом `Firecrawl` [54] та виокремлення ціни з її вмісту окремою LLM; модель `gpt-4.1-mini-2025-04-14`;

– `validate_prices(prices, normalized_query)` – обчислення оцінки впевненості для кожної знайденої ціни, перевірка синонімів і усунення дублікатів; модель `claude-haiku-4-5-20251001` (за результатами Додатку Д).

Інженерні інструменти, аналогічні підходу на основі простого агента, для звернення до зовнішніх сервісів:

– `search(query, country_code)` – вебпошук через `Tavily` [53];

– `fetch_page(url)` – завантаження сторінки через `Firecrawl` [54];

– `list_local_stores(country_code, city, radius_km)` – отримання переліку магазинів з картографічних даних `OpenStreetMap` [46] через сервіс запитів `Overpass API` [47].

Кожен LLM-інструмент використовує модель, обрану експериментально для відповідного етапу послідовності LLM, що дозволяє повторно використати оптимізаційні рішення з Додатку Д.

Великі обсяги даних, такі як тексти результатів пошуку, вміст завантажених сторінок, зберігаються в окремому контексті виконання та не передаються через діалог з агентом-оркестратором. Інструменти повертають у діалог лише короткі узагальнення з ідентифікаторами блоків, за якими наступний інструмент може звернутися до повних даних. Такий розподіл скорочує обсяг токенів, що оплачуються на кожній ітерації, та дозволяє уникнути дублювання витрат за передачу однакового вмісту до різних моделей.

2.6.3 Робочий цикл агента

Робочий цикл гібридного агента побудований за принципом, аналогічним підходу на основі простого агента: на кожній ітерації агент-оркестратор аналізує накопичену в діалозі інформацію, формулює міркування щодо наступного кроку, обирає один з доступних інструментів та подає до нього аргументи. Виклик інструменту повертає результат, який додається до контексту наступної ітерації. Цикл триває до виклику фінального інструменту `submit_final_answer` або до досягнення межі ітерацій. Узагальнену схему гібридного агента наведено у Додатку Б (рис. Б.4).

Системний промпт агента-оркестратора містить опис задачі, перелік доступних інструментів з їх сигнатурами та рекомендований типовий маршрут: нормалізація запиту, отримання переліку місцевих магазинів, вебпошук, виокремлення цін з результатів пошуку, перевірка ціни на сторінці (опційно для підозрілих результатів), скоринг та валідація, фінальна відповідь. Рекомендований маршрут є м'якою підказкою, а не обов'язковим порядком. Агент може повертатися до попередніх інструментів, повторно викликати `search` зі зміненими параметрами або пропускати кроки, якщо проміжна інформація достатня для прийняття рішення.

2.6.4 Експерименти з вибору моделі агента-оркестратора

У гібридному підході оркестратор координує спеціалізовані інструменти, а не виконує всю роботу самостійно, як у простому агенті. Тому вибір моделі-оркестратора потребує окремого підбору. Методика відбору, перелік шести моделей-кандидатів та умови тестування (температура 0, максимальна кількість ітерацій 20, набір з шести товарів, ручна розмітка) аналогічні застосованим у підпункті 2.5.4. Усього виконано 36 запусків. Моделі LLM-інструментів (підпункт 2.4.4, Додаток Д) залишались

незмінними, а змінювалась лише модель оркестратора. Повні результати порівняння наведено у Додатку И (табл. И.2).

За результатами обрано модель gpt-5-mini, ту саму, що й для простого агента. Вона показала точність 0,71, чесність самооцінки 0,80, правильну найдешевшу ціну у 83 % товарів, середню кількість правильно знайдених цін на запуск 6,2, середню вартість одного запуску 0,096 долара та середній час 170 секунд. Моделі з вищою або рівною точністю, такі як gpt-5.1 (0,76) та gemini-2.5-pro (0,73), мають вартість у 1,5–2,2 рази вищу (0,141 та 0,211 долара відповідно). Модель gemini-3-flash-preview виявилась найгіршою через низьку чесність самооцінки (0,50) та надмірний час одного запуску (507 секунд).

Використання однакової моделі для обох агентних підходів забезпечує методологічну коректність порівняння, оскільки будь-які розбіжності у показниках обумовлені відмінностями архітектур, а не вибором моделі.

2.6.5 Експериментальна перевірка підходу

Гібридний агент випробувано на всіх 20 позиціях набору даних (табл. А.1). Релевантність кожного знайденого запису визначено вручну за критеріями підрозділу 2.1. Детальні результати за окремими позиціями наведено у Додатку В (табл. В.4).

На відміну від простого агента (підрозділ 2.5), гібридний агент повернув знахідки для всіх 20 позицій. Зокрема, п'ять позицій, на яких простий агент вичерпав ліміт ітерацій без знахідок («Losos 4×125g», «Pilos Tvar. 0,5 %», «LINDT ALKOHOL EIER SORTIE», «Bonduelle Bio Mais», «MANDARINE SPAR GELEG»), гібридний агент обробив успішно з точністю від 50 % до 88 %. Делегування спеціалізованих задач окремим LLM-інструментам дозволяє оркестратору ефективніше витрачати ітерації на

пошук, а не на обробку вмісту сторінок. Якісний аналіз результатів виявив три типові класи поведінки.

Висока точність спостерігається для позицій, де більшість знахідок релевантні, зокрема «Bounty 57g» (100 %), «Батон 0,5 кг Кулиничі Дорожній» (100 %), «Par. San Marzano» (89 %), «Losos 4×125g» (88 %), «Напій 1 л Sprite ПЕТ» (88 %), «Twin. Engl.Breakf.» (88 %), «Сіль 1 кг Вибір кам'яна мелена» (88 %). На таких товарах гібридний агент поєднує переваги обох попередніх підходів: адаптивну стратегію пошуку та спеціалізовану обробку даних.

Зниження точності через варіанти – позиції, де у видачу потрапляють інші варіанти або нерелевантні записи, зокрема «Lktf. mlieko» (50 %), «Томати коктейльні 1 гат» (50 %), «MANDARINE SPAR GELEG» (50 %), «COKE ZERO ZERO 0,33L» (38 %). Наприклад, для «Lktf. mlieko» найдешевшою виявилася нерелевантна знахідка (інший тип молока за 0,39), а для «COKE ZERO ZERO 0,33L» серед знахідок переважають варіанти звичайної Coca-Cola.

Правильний товар із непідтвердженою ціною є окремим класом помилок, характерним для гібридного агента. На позиціях «Томати коктейльні 1 гат» та «Морозиво 0,6 кг Ласунка «Mix Deluxe»» найдешевший запис є точною відповідністю шуканому товару, проте зазначена ціна не підтверджена на сторінці-джерелі. Агент правильно ідентифікує товар, але етап перевірки ціни не завжди виявляє розбіжність між ціною у видачі та фактичною ціною на сторінці.

Загалом гібридний агент повертає знахідки для всіх позицій та обробляє їх швидше за простий агент, проте з нижчою часткою релевантних записів серед знайдених. Делегування спеціалізованих задач (нормалізація, виокремлення цін, валідація) окремим LLM-інструментам зменшує навантаження на оркестратор, але водночас вносить додаткові джерела похибок, зокрема на етапі перевірки актуальності цін.

2.6.6 Висновки щодо точності, часових витрат та вартості

Точність. На всіх 20 позиціях гібридний агент повернув 149 записів із середньою точністю 74 % та строгою точністю 60 %. Охоплення становить 5,2 релевантного магазину на товар. Правильну найдешевшу ціну підхід повернув для 55 % товарів. Для жодної позиції не залишився без знахідок. Чесність самооцінки становить 71 %, що нижче, ніж у простого агента (88 %), що свідчить про менш точну внутрішню оцінку збігів у гібридній архітектурі.

Часові витрати. Середній час обробки однієї позиції становить 203 секунди, сумарний час прогону по 20 позиціях – близько 68 хвилин.

Вартість. Середня вартість обробки однієї позиції становить 0,114 долара, що є найнижчою серед трьох підходів на основі LLM (порівняно з 0,146 у послідовності та 0,185 у простому агенті). Сумарна вартість прогону по 20 позиціях складає 2,28 долара. Зниження вартості порівняно з простим агентом пояснюється тим, що спеціалізовані LLM-інструменти обробляють дані поза контекстом оркестратора, уникаючи дублювання витрат на токени.

Порівняно з простим агентом (підрозділ 2.5) гібридний агент забезпечує знахідки для всіх 20 позицій (проти 15 у простого), швидшу обробку (203 проти 335 секунд) та нижчу вартість (0,114 проти 0,185 долара). Натомість він має нижчу точність (74 % проти 88 %), строгу точність (60 % проти 71 %) та правильну найдешевшу ціну (55 % проти 67 %). Порівняно з послідовністю LLM (підрозділ 2.4) гібридний агент має вищу точність (74 % проти 57 %) і строгу точність (60 % проти 39 %), нижчу вартість (0,114 проти 0,146 долара), але менше охоплення (5,2 проти 7,1 магазину) та довший час обробки (203 проти 119 секунд).

Гібридний підхід поєднує сильні сторони послідовності LLM (висока якість обробки на спеціалізованих задачах за рахунок моделей, підібраних для кожного етапу) та простого агента (гнучкість послідовності дій, адаптивність до нестандартних товарів).

Серед сильних сторін підходу:

- знахідки для всіх позицій без вичерпання ліміту ітерацій;
- найнижча вартість серед підходів на основі LLM (0,114 долара на позицію) завдяки обробці даних поза контекстом оркестратора;
- швидша обробка порівняно з простим агентом (203 проти 335 секунд) завдяки делегуванню обробки даних.

Серед обмежень:

- нижча точність порівняно з простим агентом (74 % проти 88 %), оскільки делегування обробки окремим інструментам вносить додаткові джерела похибок;
- нижча чесність самооцінки (71 % проти 88 %);
- правильна найдешевша ціна лише для 55 % товарів, на рівні послідовності LLM;
- проблема непідтвердженої ціни, коли агент правильно ідентифікує товар, але етап перевірки не завжди виявляє розбіжність між ціною у видачі та фактичною ціною на сторінці.

2.7 Розробка алгоритму моніторингу цін товарів з чеків

2.7.1 Порівняння розглянутих підходів

У підрозділах 2.3–2.6 розглянуто чотири підходи до автоматизованого пошуку цін на товари з касових чеків. Усі підходи випробувано на єдиному наборі з 20 позицій за єдиними метриками (визначення метрик наведено у підрозділі 2.1). Зведені результати наведено у таблиці 2.2.

Жоден з чотирьох підходів не домінує за всіма метриками одночасно. Підхід на основі вебскрейпінгу (підрозділ 2.3) не потребує грошових витрат на зовнішні сервіси, проте має найнижчу точність (50 %), найвужче охоплення (1,6 магазину на товар) та не є універсальним по країнах і мовах.

Таблиця 2.2 – Порівняння розглянутих підходів за ключовими характеристиками

Характеристика	Вебскрейпінг	Послідовність LLM	Простий агент	Гібридний агент
Точність	50 %	57 %	88 %	74 %
Строга точність	39 %	39 %	71 %	60 %
Охоплення (магазинів на товар)	1,6	7,1	5,3	5,2
Мін. ціна вірна	47 %	55 %	67 %	55 %
Чесність самооцінки	-	53 %	88 %	71 %
Товарів без знахідок	1	0	5	0
Середній час на позицію, с	369	119	335	203
Середня вартість на позицію, \$	0	0,146	0,185	0,114
Універсальність по країнах	потребує адаптації	універсальний	універсальний	універсальний
Адаптивність до товару	низька (правила)	низька (фікс. потік)	висока (адапт. цикл)	висока (адапт. цикл)
Прозорість процесу	висока (код)	висока (фікс. потік)	низька (LLM)	середня (LLM + етапи)

Послідовність LLM (підрозділ 2.4) забезпечує найширше охоплення (7,1 магазину на товар) та найшвидшу обробку (119 секунд), проте має низьку точність (57 %) і найнижчу чесність самооцінки (53 %) серед LLM-підходів.

Простий агент (підрозділ 2.5) демонструє найвищу точність (88 %), строгу точність (71 %) та чесність самооцінки (88 %). Натомість він має

найбільшу кількість товарів без знахідок (5 з 20), найвищу вартість (0,185 долара на позицію) та значний час обробки (335 секунд).

Гібридний агент (підрозділ 2.6) займає проміжну позицію: точність (74 %) і строга точність (60 %) вищі за послідовність, але нижчі за простий агент. Водночас він повертає знахідки для всіх 20 позицій, має найнижчу вартість серед LLM-підходів (0,114 долара) та помірний час обробки (203 секунди).

Задача моніторингу цін не потребує відповіді в реальному часі, тому пріоритет надано надійності та якості над швидкістю обробки. Для впровадження у мобільний застосунок обрано підхід на основі гібридного агента з огляду на такі міркування:

- надійність – підхід повертає знахідки для всіх позицій; порожня відповідь на кожний четвертий товар, характерна для простого агента, є неприйнятною для користувацького досвіду;

- вартість – 0,114 долара на позицію є найнижчою серед LLM-підходів, що на 38 % дешевше за простий агент та на 22 % дешевше за послідовність;

- точність – 74 % нижча за простий агент (88 %), проте суттєво вища за послідовність (57 %) та вебскрейпінг (50 %);

- час обробки – 203 секунди на позицію; повільніше за послідовність (119 секунд), але на 39 % швидше за простий агент (335 секунд);

- масштабованість – спеціалізовані LLM-інструменти можна вдосконалювати окремо (замінити модель або промпт на одному етапі без впливу на решту), що спрощує підтримку у виробничій експлуатації.

2.7.2 Алгоритм моніторингу цін на основі гібридного агента

На підставі порівняння (підпункт 2.7.1) обрано гібридний агент як основу модуля пошуку цін. Алгоритм роботи сервісу моніторингу складається з трьох етапів.

На першому етапі користувач завантажує фотографію касового чека у мобільний застосунок. Система розпізнавання виокремлює товарні рядки та для кожного формує вхідні дані, а саме назву позиції з чека, місто та код країни. Розпізнавання вмісту чека виконується наявними засобами застосунку і не є предметом цієї роботи.

На другому етапі для кожної обраної користувачем позиції запускається гібридний агент (підрозділ 2.6). Агент-оркестратор gpt-5-mini послідовно викликає спеціалізовані інструменти: нормалізує запит, шукає ціни через вебпошук Tavily, завантажує сторінки магазинів через Firecrawl, виокремлює та валідує знахідки. Результатом є структурований перелік цін із зазначенням магазину, ціни та оцінки відповідності. Середня вартість обробки однієї позиції становить 0,114 долара, середній час – 203 секунди.

На третьому етапі знахідки подаються користувачеві у мобільному інтерфейсі, розподілені за категоріями відповідності. Для точних відповідностей виокремлено найдешевшу пропозицію як рекомендовану. Користувач може переглянути всі знайдені варіанти та перейти на сторінку магазину для підтвердження ціни.

3 РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ МОНІТОРИНГУ ЦІН НА ТОВАРИ З ПАПЕРОВОГО ЧЕКА

3.1 Технічне завдання

Модуль моніторингу цін розроблено як Python-бібліотеку, призначену для інтеграції у мобільні застосунки для відстеження витрат. Модуль приймає назву товарної позиції з касового чека та інформацію про місцезнаходження покупки, виконує автоматизований пошук цін на цей товар у торговельних мережах регіону та повертає структурований перелік знахідок із зазначенням магазину, ціни та оцінки відповідності.

Функціональні вимоги до модуля визначають формати вхідних і вихідних даних. На вхід модуль приймає три обов'язкові параметри: назву позиції з касового чека (текстовий рядок в оригінальному вигляді, без попередньої обробки), код країни покупки (двосимвольний код ISO 3166-1, наприклад «SK», «UA», «AT») та назву міста покупки. Додатково може бути передана назва магазину-джерела (мережа, де здійснено покупку), щоб виключити його з результатів пошуку.

На виході модуль повертає структурований результат. Він містить нормалізований опис шуканого товару (бренд, тип продукту, підтип, об'єм або вага, додаткові атрибути, повна читабельна назва) та статус обробки (успішний пошук, приватна марка, відсутність знахідок або неоднозначний запит). Основною складовою є перелік знайдених цінових пропозицій, відсортований за зростанням ціни, де кожна знахідка містить назву магазину, назву товару на сайті магазину, ціну, валюту, посилання на сторінку-джерело, ознаку акційної ціни та оцінку відповідності з поясненням. Окремим полем виділено найдешевшу точну відповідність для швидкого доступу. Також результат містить журнал роботи агента (послідовність кроків з описом міркувань) та сумарні витрати на обробку запиту (кількість токенів, вартість у доларах).

Нефункціональні вимоги до модуля враховують особливості цільового використання та визначають кількісні пороги якості:

- час обробки одного запиту не повинен перевищувати п'яти хвилин, оскільки задача моніторингу цін не потребує відповіді в реальному часі, модуль працює у фоновому режимі, а користувач переглядає результати після завершення обробки;

- точність знайдених записів має становити не менше 70 %, тобто щонайменше сім із десяти повернутих знахідок мають бути релевантними шуканому товару;

- охоплення має становити не менше чотирьох різних магазинів на товар, що забезпечує користувачеві достатню кількість альтернатив для порівняння;

- повнота – модуль має повернути хоча б одну знахідку для не менше ніж 90 % оброблених товарів;

- вартість обробки одного запиту не повинна перевищувати 0,20 долара, що забезпечує економічну доцільність масового використання.

Модуль не зберігає і не передає персональні дані користувача. На вхід подаються лише назва товару, код країни та місто, і жодна з цих величин не містить інформації, що дає змогу ідентифікувати особу. Це спрощує дотримання вимог законодавства про захист персональних даних при розгортанні в різних юрисдикціях.

Модуль має бути доступний для виклику цілодобово, оскільки користувачі мобільного застосунку можуть ініціювати пошук у довільний час. Забезпечення безперебійної роботи покладено на серверну інфраструктуру, до якої підключається модуль. Сам модуль не зберігає проміжного стану між викликами, він приймає запит, обробляє його та повертає результат, не потребуючи власного сервера та бази даних.

3.2 Налаштування програмного середовища

Модуль моніторингу цін реалізовано мовою Python версії 3.11. Вибір мови обумовлено наявністю офіційних клієнтських бібліотек усіх задіяних провайдерів LLM (OpenAI, Google, Anthropic), а також сумісністю з існуючою серверною інфраструктурою цільового застосунку.

Модуль оформлено як Python-пакет зі стандартною структурою збирання на основі hatchling. Для встановлення достатньо виконати стандартну команду `pip install`, що автоматично встановлює усі залежності. Перелік основних залежностей із зазначенням мінімальних версій наведено далі.

Бібліотека `google-generativeai` версії 1.0 забезпечує взаємодію з API Google Gemini, який використовується для агента-оркестратора. Бібліотека `openai` версії 1.40 забезпечує взаємодію з API OpenAI, через який викликаються моделі для спеціалізованих LLM-інструментів (нормалізація, виокремлення цін, перевірка). Бібліотека `anthropic` версії 0.40 забезпечує взаємодію з API Anthropic, через який викликається модель для валідації знахідок.

Бібліотека `tavily-python` версії 0.5 реалізує клієнт сервісу Tavily Search [53], що використовується як інструмент вебпошуку агента. Бібліотека `requests` версії 2.31 використовується для HTTP-запитів до сервісу Firecrawl (завантаження сторінок) [54] та до Overpass API (пошук торговельних точок через OpenStreetMap [46]).

Бібліотека `pydantic` версії 2.0 забезпечує валідацію та серіалізацію структурованих даних, а саме вхідних параметрів, відповідей LLM та результатів модуля. Бібліотека `python-dotenv` версії 1.0 використовується для завантаження конфігураційних змінних (ключі API, параметри моделей) із файлу оточення.

Модуль взаємодіє з п'ятьма зовнішніми сервісами через їхні API. Tavily Search API виконує вебпошук за текстовими запитам та повертає до 20 результатів із фрагментами вмісту сторінок. Firecrawl API завантажує

вебсторінки з відтворенням динамічного вмісту та обходом захисних механізмів (зокрема Cloudflare), повертаючи вміст у форматі Markdown. Overpass API (OpenStreetMap) виконує пошук торговельних мереж у радіусі 15 кілометрів від зазначеного міста. API трьох провайдерів LLM (OpenAI, Google Gemini, Anthropic) забезпечують виклик моделей для оркестрації та спеціалізованих інструментів.

Модуль не потребує бази даних, вебсервера або фронтенд-компонентів. Усі проміжні дані зберігаються в оперативній пам'яті протягом обробки одного запиту та звільняються після повернення результату. Конкретні моделі, задіяні на кожному етапі обробки, та їхні параметри розглянуто у наступному підпункті.

3.3 Особливості підключення та налаштування LLM та зовнішніх сервісів

Модуль одночасно використовує моделі трьох провайдерів: OpenAI, Google Gemini та Anthropic. Маршрутизація між провайдерами виконується автоматично на підставі префікса назви моделі: «gpt-» спрямовує виклик до OpenAI, «gemini-» – до Google Gemini, «claude-» – до Anthropic. Це дає змогу замінити модель на будь-якому етапі зміною одного рядка конфігурації без модифікації коду модуля.

Конкретні моделі, задіяні на кожному етапі обробки, наведено далі. Агент-оркестратор використовує модель gpt-5-mini (OpenAI), обрану за результатами експериментального порівняння (підпункт 2.6.4). Спеціалізовані LLM-інструменти нормалізації, виокремлення цін та перевірки використовують модель gpt-4.1-mini-2025-04-14 (OpenAI), обрану за результатами порівняння моделей для послідовності LLM (підрозділ 2.4). Інструмент валідації використовує модель claude-haiku-4-5-20251001 (Anthropic), обрану за тими самими результатами.

Усі виклики LLM виконуються з температурою генерації 0. Нульова температура забезпечує детерміновану поведінку моделей: за однакового вхідного тексту модель повертає однаковий результат, що підвищує відтворюваність і зменшує варіативність якості між запусками.

Відповіді спеціалізованих LLM-інструментів обмежено структурованим форматом на основі заздалегідь визначених Pydantic-схем [55]. Кожен інструмент має власну схему відповіді. Нормалізація повертає розібрану назву товару з переліком пошукових запитів, виокремлення – перелік цінових пропозицій із зазначенням магазину та URL, перевірка – підтвердження ціни на конкретній сторінці, валідація – оцінку впевненості для кожної знахідки. Структурований формат усуває потребу розбирати довільний текст і зменшує кількість помилок при передачі даних між етапами.

Ліміт ітерацій оркестратора встановлено на рівні 30 інструментних викликів на один запит. Це значення визначено емпірично: робочий цикл агента передбачає послідовну перевірку кожної знайденої ціни на сторінці-джерелі, що додає один крок на кожну знахідку. При типовій кількості 6–9 знахідок на товар цикл нормалізація, пошук, виокремлення, перевірка, валідація, відповідь вкладається у 15–25 кроків, а ліміт 30 залишає запас для повторних пошуків у складних випадках.

Для забезпечення надійності реалізовано механізм автоматичного переключення моделі оркестратора. Якщо основна модель повертає помилку (наприклад, перевантаження серверів, код 503), система автоматично повторює виклик із резервною моделлю. Кожний виклик фіксує фактично використану модель у журналі витрат, що дає змогу відстежити, які запити оброблено основною, а які – резервною моделлю.

Зовнішні сервіси пошуку та завантаження сторінок також потребують налаштування. Для сервісу вебпошуку Tavily встановлено ліміт 20 результатів на один запит та таймаут 30 секунд. Для сервісу завантаження сторінок Firecrawl встановлено таймаут 40 секунд на одну сторінку та час

очікування завантаження динамічного вмісту 2 500 мілісекунд. Ці значення визначено емпірично: робочі сторінки торговельних мереж завантажуються за 6–12 секунд, а таймаут 40 секунд відсікає сторінки, що не відповідають або не завантажуються. Для сервісу пошуку торговельних точок через Overpass API (OpenStreetMap) встановлено радіус пошуку 15 кілометрів та таймаут 25 секунд.

3.4 Розробка архітектури модуля

Внутрішня архітектура модуля моніторингу цін побудована навколо чотирьох компонентів: точки входу, агентного циклу, спільного робочого простору та набору інструментів. Діаграму компонентів модуля з потоками даних наведено на рисунку 3.1.

Точка входу реалізована функцією `find_prices_for_item`, яка приймає вхідні параметри (підпункт 3.1), створює робочий простір, запускає агентний цикл та після його завершення збирає результат. На цьому рівні агрегуються сумарні витрати на токени та виклики зовнішніх сервісів, формується перелік знахідок, визначається найдешевша точна відповідність та повертається структурований результат.

Агентний цикл реалізує оркестрацію за патерном ReAct [44]. На кожній ітерації модель-оркестратор аналізує поточний стан діалогу, обирає наступний інструмент та формує параметри його виклику. Диспетчер виконує обраний інструмент, записує результат у робочий простір та повертає оркестратору стислу довідку про виконану дію. Цикл завершується, коли оркестратор викликає фінальний інструмент `submit_final_answer` або коли досягнуто ліміт ітерацій (30 кроків). У разі вичерпання ліміту модуль не повертає порожній результат, а збирає знахідки, вже накопичені в робочому просторі на попередніх кроках, що підвищує надійність.

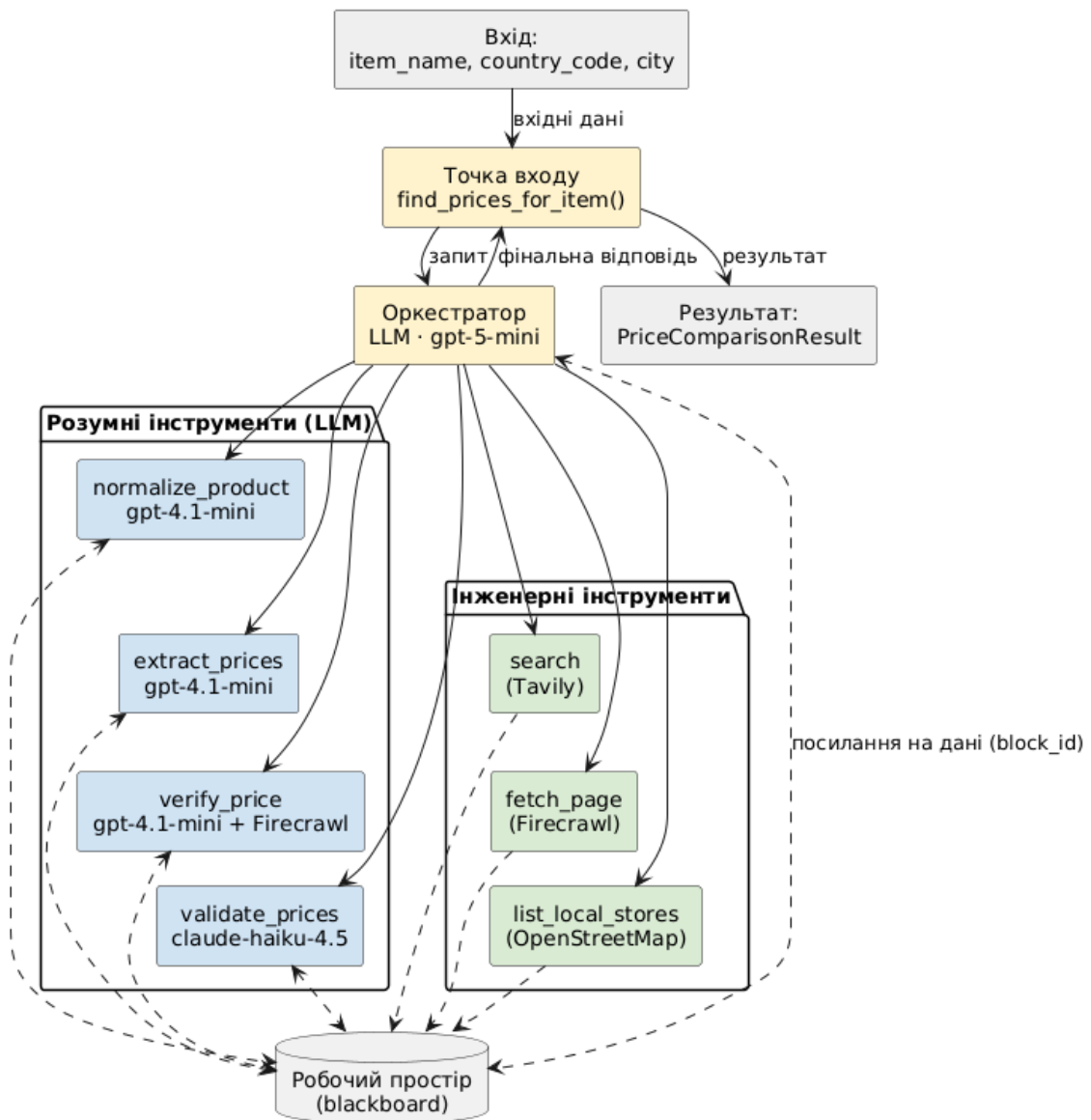


Рисунок 3.1 – Діаграма компонентів модуля моніторингу цін

Взаємодія з моделлю-оркестратором побудована як багатокроковий діалог. Кожна ітерація циклу відповідає одному ходу діалогу. Модуль передає оркестратору результат попереднього інструмента, оркестратор аналізує накопичену історію та повертає виклик наступного інструмента. Історія діалогу зростає з кожним кроком, і оркестратор бачить усі свої попередні рішення та результати, що дає йому змогу враховувати контекст при плануванні наступних дій. Порядок інструментів у переліку декларацій (нормалізація, пошук магазинів, вебпошук, виокремлення, перевірка, оцінювання) слугує неявною підказкою для моделі. Оркестратор схильний

дотримуватись цього порядку на початку роботи, але може відхилятися від нього на підставі проміжних результатів.

Спільний робочий простір реалізує патерн «дошка» (англ. blackboard) [56]. У класичному патерні керування здійснює детермінований планувальник на основі правил. У розробленому модулі цю роль виконує LLM-оркестратор, який приймає рішення на підставі контексту діалогу, а не фіксованих правил. Усі інструменти записують проміжні результати в єдину структуру даних, доступну на читання і запис кожному наступному інструменту. Оркестратор не бачить цих масивів даних безпосередньо, він отримує лише стислі довідки від інструментів. Таке рішення має дві переваги. По-перше, великі обсяги тексту (сторінки магазинів, результати пошуку) не передаються через контекст оркестратора, що зменшує витрати на токени. По-друге, інструменти можуть накопичувати та усувати дублікати між послідовними викликами без участі оркестратора.

Робочий простір містить такі структуровані поля: нормалізований опис товару (бренд, тип, підтип, об'єм, пошукові запити та ключові слова варіантів), блоки результатів вебпошуку (повний текст кожного пошуку, індексований за ідентифікатором блоку), таблицю виокремлених цін (магазин, назва товару, ціна, валюта, URL-джерело), перелік локальних торговельних мереж та результати оцінювання відповідності (оцінка впевненості для кожної знахідки). Дані у робочому просторі зростають від кроку до кроку. Після нормалізації заповнюється опис товару, після кожного пошуку додається новий блок тексту, після виокремлення зростає таблиця цін, після перевірки оновлюються ціни окремих записів, після оцінювання додаються оцінки впевненості.

Набір інструментів складається з семи одиниць та одного фінального інструмента. Чотири спеціалізовані LLM-інструменти виконують етапи обробки даних, а саме нормалізацію запиту з чека, виокремлення цін із результатів пошуку, перевірку ціни на сторінці магазину та оцінювання відповідності знахідок із визначенням оцінки впевненості. Три інженерні

інструменти звертаються до зовнішніх сервісів: вебпошук через Tavily, завантаження сторінок через Firecrawl та пошук торговельних точок через OpenStreetMap. Фінальний інструмент `submit_final_answer` сигналізує про завершення роботи та передає оркестратору структурований перелік знахідок для формування відповіді.

Типову послідовність взаємодії оркестратора з інструментами під час обробки одного товару наведено на рисунку 3.2.

За даними тестування на 20 позиціях (підрозділ 2.6), на один товар у середньому припадає 19,6 кроків: 1 нормалізація, 1 пошук магазинів, 5,8 вебпошуків (від 2 до 12), 2,1 виокремлення цін, 7,5 перевірок ціни (від 5 до 12) та 1 оцінювання відповідності. На відміну від фіксованої послідовності, агент може повертатися до пошуку та виокремлення після перевірки цін, якщо оцінює охоплення магазинів як недостатнє. Наприклад, для товару «Мієко Рајо Безл 1,5 % 1l» агент виконав три цикли пошуку з проміжною перевіркою знахідок, після чого знайшов та перевіряв додатково ще два записи.

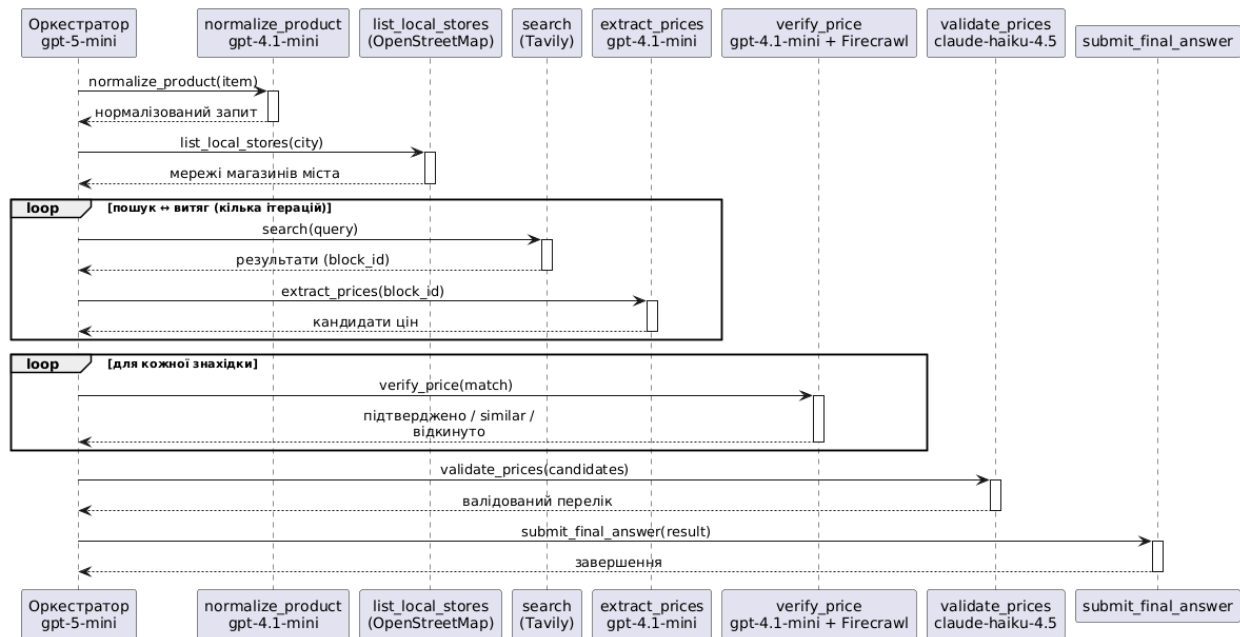


Рисунок 3.2 – Діаграма послідовності обробки одного товару

Кожен крок агентного циклу записується до журналу виконання, що зберігається у вихідному результаті модуля. Журнал містить номер кроку, опис міркувань оркестратора, назву викликаного інструмента, його параметри та автоматично сформований опис результату. Це забезпечує прозорість роботи модуля та спрощує діагностику помилок.

3.5 Опис мобільного застосунку Billka AI та можливості впровадження модуля

Billka AI [33, 34] – мобільний застосунок для обліку витрат, доступний у Google Play та App Store. Основна функціональність застосунку полягає у скануванні касових чеків, розпізнаванні товарних позицій та агрегуванні витрат за категоріями. Внутрішня архітектура Billka AI не є предметом цієї роботи, тому застосунок розглядається як хост-система, до якої підключається розроблений модуль моніторингу цін.

Модуль інтегрується у серверну частину Billka AI як Python-бібліотека, що викликається у тому ж процесі (рис. 3.3). Це рішення прийнято за побажанням команди розробки застосунку: бібліотечний виклик простіший за розгортання окремого мікросервісу, не потребує мережевої взаємодії між модулем та хост-системою і не додає затримок на серіалізацію запитів. Єдині HTTPS-з'єднання, які створює модуль, спрямовані до зовнішніх сервісів: Tavily (вебпошук), Firecrawl (завантаження сторінок), OpenStreetMap (пошук магазинів) та API провайдерів LLM (OpenAI, Anthropic, Google Gemini).

Точкою з'єднання між застосунком та модулем є функція `find_prices_for_item`. Застосунок формує виклик, передаючи назву товарної позиції з розпізнаного чека, код країни, місто та за потреби назву магазину-джерела. Модуль виконує пошук цін у фоновому режимі та повертає структурований результат, тобто перелік знахідок із зазначенням магазину, ціни та категорії відповідності. Можливі категорії відповідності: точна

відповідність, схожа знахідка та інший варіант (у програмному коді модуля позначаються як `exact`, `similar` та `different_variant` відповідно). Застосунок відображає ці дані користувачеві, групуючи знахідки за категорією відповідності та виокремлюючи найдешевшу точну відповідність.

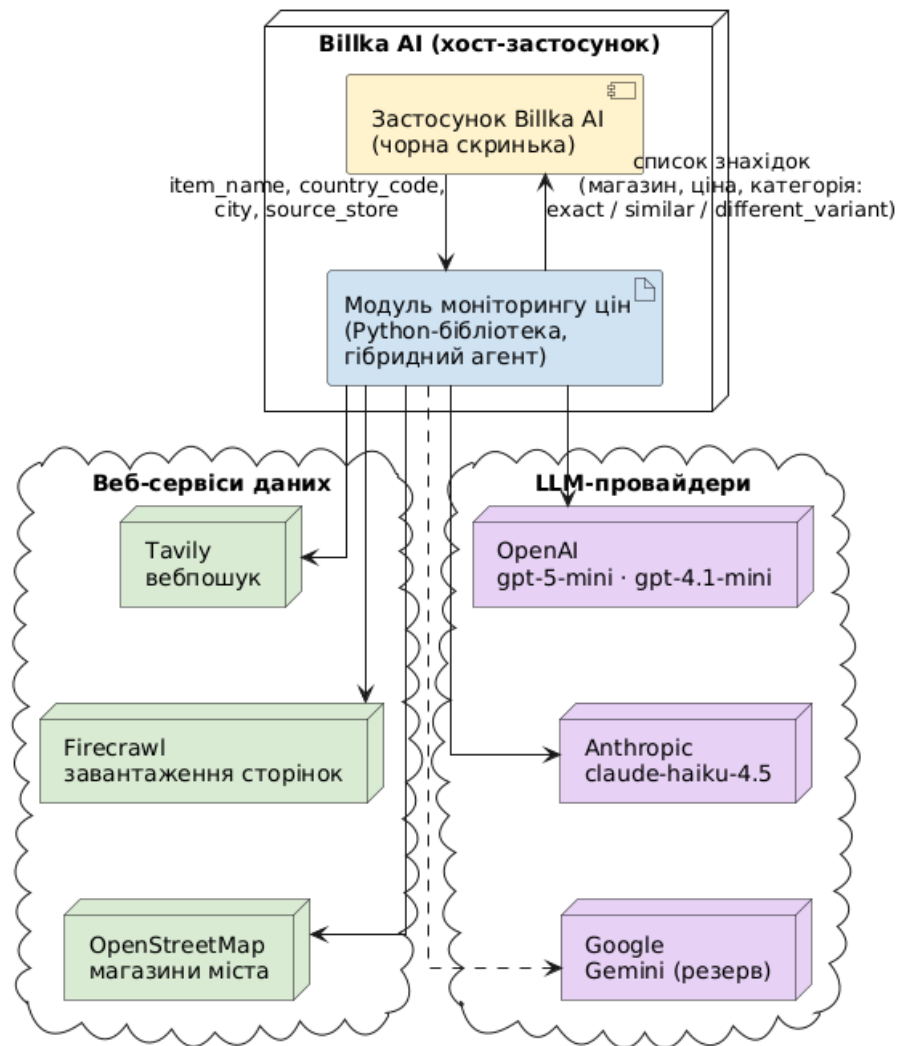


Рисунок 3.3 – Діаграма розгортання модуля у складі застосунку Billka AI

Модуль не зберігає стану між викликами та не має доступу до бази даних застосунку. Якщо потрібно реалізувати відстеження цін на товар у часі (історія моніторингу), логіку повторних запусків та збереження результатів покладено на застосунок Billka AI. Модуль виконує лише одноразовий пошук за запитом і повертає результат.

3.6 Розробка модуля моніторингу цін

У підпункті 3.4 описано загальну архітектуру модуля та структуру робочого простору. Цей підпункт розкриває ключові програмні рішення, прийняті при реалізації.

Взаємодія оркестратора з інструментами реалізована через механізм виклику функцій (англ. function calling), вбудований у пакети засобів розробки (англ. Software Development Kit, SDK) провайдерів LLM. Кожен інструмент описаний декларацією у форматі JSON Schema, що містить назву, опис призначення та перелік параметрів із типами. Набір декларацій передається моделі-оркестратору при ініціалізації сесії. На кожному кроці оркестратор повертає не довільний текст, а структурований виклик конкретного інструмента з заповненими параметрами. Диспетчер модуля зіставляє назву виклику з відповідною функцією та передає їй параметри. Кожна функція-інструмент при створенні прив'язується до робочого простору (підпункт 3.4) через механізм часткового застосування (англ. partial application), тому оркестратор передає лише легковагові аргументи, а важкі дані інструмент читає та пише безпосередньо з робочого простору.

Відповіді спеціалізованих LLM-інструментів обмежено структурованим форматом на основі заздалегідь визначених Pydantic-схем. Реалізація структурованого виведення відрізняється залежно від провайдера LLM. Для моделей Google Gemini використовується параметр `response_schema` із зазначенням типу відповіді `application/json`: модель генерує лише валідний JSON, що відповідає заданій схемі. Для моделей OpenAI використовується параметр `response_format` із передачею Pydantic-моделі, після чого SDK автоматично перетворює схему у JSON Schema та валідує відповідь. Для моделей Anthropic структуроване виведення реалізовано через механізм `tool_use` із примусовим вибором інструмента, де модель повертає відповідь як аргументи виклику фіктивного інструмента, що

гарантує відповідність заданій схемі. Ці відмінності інкапсульовано в одній функції маршрутизації, тому решта коду модуля не залежить від провайдера.

Діаграму активності агента з розгалуженнями та циклами наведено на рисунку 3.4.

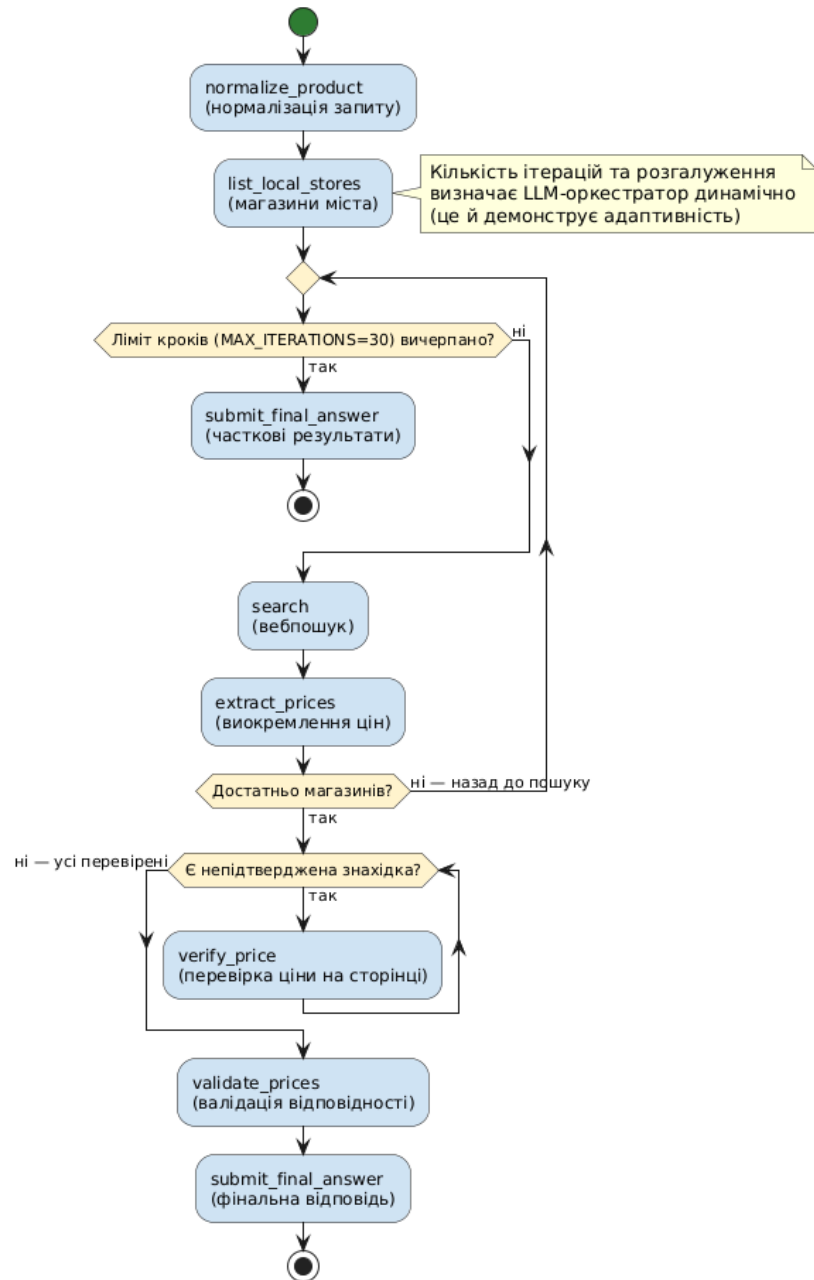


Рисунок 3.4 – Діаграма активності гібридного агента

Системний промпт оркестратора визначає рекомендовану стратегію роботи агента. Промпт містить покроковий опис робочого процесу:

нормалізація (один раз, першою), пошук магазинів, серія вебпошуків із різними формулюваннями (локальна мова, англійська, агрегатори, платформи доставки), виокремлення цін, перевірка кожної знахідки на сторінці магазину, оцінювання відповідності та фінальна відповідь. Водночас промпт явно зазначає, що оркестратор може адаптувати послідовність, наприклад, повертатися до пошуку для розширення охоплення, додавати цільові запити на основі виявлених агрегаторів або ще не охоплених мереж. Критерії завершення пошуку також закладені у промпті. Агент зупиняється, коли знайдено щонайменше 8 різних магазинів, або коли два останні виокремлення не додали нових магазинів, або коли наближається ліміт ітерацій. В останньому випадку промпт наказує негайно перевірити, оцінити та подати наявні знахідки, не витрачаючи залишок кроків на нові пошуки.

Кожен спеціалізований LLM-інструмент має власний промпт та схему відповіді. Інструмент нормалізації спрямовує модель на розбір скороченої назви з чека у структуровані поля (бренд, тип, підтип, об'єм, додаткові атрибути) та генерацію 6–8 пошукових запитів різними мовами. Додатково генеруються переліки ключових слів для розпізнавання варіантів та мультипаків, що використовуються на наступних етапах. Результат записується у робочий простір та доступний усім наступним інструментам.

Інструмент виокремлення цін аналізує повні тексти результатів вебпошуку, збережені у робочому просторі, та повертає перелік знахідок із зазначенням магазину, назви товару, ціни, валюти та URL-джерела. При повторних викликах нові знахідки додаються до єдиної таблиці цін у робочому просторі з автоматичним усуненням дублікатів. Запис із тією самою назвою магазину та назвою товару не додається повторно.

Інструмент перевірки ціни завантажує конкретну сторінку магазину через Firecrawl та спрямовує модель на повторне виокремлення ціни безпосередньо з вмісту сторінки. Це дає змогу виявити розбіжності між ціною у пошуковій видачі та фактичною ціною на сторінці магазину. За результатом перевірки ціна у робочому просторі оновлюється.

Інструмент оцінювання відповідності оцінює кожну накопичену знахідку за чотирма критеріями (бренд, підтип, об'єм, додаткові атрибути) та присвоює оцінку впевненості від 0 до 1. Для цього інструменту обрано модель іншого провайдера (claude-haiku-4-5 замість gpt-4.1-mini), що зменшує ризик систематичної похибки при оцінюванні результатів, згенерованих моделями того самого провайдера на попередніх етапах. Оцінка впевненості відображається на категорії відповідності. Значення 0,8 і вище означає точну відповідність, від 0,5 до 0,8 позначає схожу знахідку, нижче 0,5 – інший варіант.

Обробка помилок реалізована на кількох рівнях. На рівні інструментів кожний виклик обгорнуто блоком перехоплення винятків. Якщо інструмент отримує невалідні аргументи від оркестратора (наприклад, відсутній обов'язковий параметр), модуль повертає оркестратору повідомлення про помилку замість аварійного завершення. Оркестратор бачить це повідомлення у контексті діалогу та може скоригувати виклик на наступному кроці. На рівні зовнішніх сервісів виклики до Tavily та Firecrawl також обгорнуто перехопленням. Якщо сервіс не відповідає протягом таймауту або повертає помилку, інструмент повертає порожній результат, і оркестратор може спробувати альтернативний запит. На рівні LLM якщо модель повертає невалідний JSON при структурованому виведенні, виконується повторна спроба. Якщо основна модель оркестратора недоступна (помилка 503 або перевантаження), система автоматично переключається на резервну модель зі списку.

Підрахунок витрат ведеться протягом усієї обробки запиту. Для кожного виклику LLM фіксується кількість вхідних, вихідних та кешованих токенів, а вартість обчислюється за тарифною таблицею провайдера. Тарифна таблиця зберігається у конфігурації модуля та містить ціни за мільйон токенів для кожної моделі, окремо для вхідних, кешованих та вихідних. Наприклад, для моделі gpt-5-mini тарифи становлять 0,25 / 0,025 / 2,00 долара за мільйон токенів відповідно. Окремо враховуються витрати на

зовнішні сервіси: 0,008 долара за один виклик Tavily та 0,00083 долара за один кредит Firecrawl. Провайдери LLM підтримують механізм автоматичного кешування (англ. *implicit caching*). При повторних викликах у межах однієї сесії системний промпт та початкові повідомлення діалогу не обробляються повторно, а зчитуються з кешу за зниженим тарифом. Наприклад, для моделі gpt-5-mini кешований вхідний токен коштує 0,025 долара за мільйон, що у 10 разів менше за звичайний вхідний (0,25 долара). Оскільки агентний цикл виконує 15–25 викликів оркестратора з однаковим системним промптом, кешування суттєво зменшує сумарну вартість. Перший виклик оплачує повну ціну за промпт, а всі наступні – лише за нові повідомлення. Модуль фіксує кількість кешованих токенів та обчислену економію у вихідному результаті. Сумарні витрати повертаються у вихідному результаті модуля, що дає змогу контролювати вартість обробки на рівні окремих запитів.

Для підвищення надійності реалізовано механізм збереження часткових результатів. Якщо оркестратор вичерпує ліміт ітерацій до виклику фінального інструмента, модуль не повертає порожню відповідь. Замість цього він збирає знахідки, вже накопичені у робочому просторі. Якщо етап оцінювання відповідності було завершено, використовуються оцінені записи з оцінками впевненості. В іншому випадку використовуються попередньо виокремлені ціни без оцінювання. Такий підхід гарантує, що навіть у складних випадках користувач отримує результат.

Окремої уваги заслуговує обробка крайніх випадків, характерних для задачі моніторингу цін. Позиції з чеків приватних марок торговельних мереж (наприклад, Picos у Lidl, S-Budget у SPAR) не мають аналогів в інших магазинах за визначенням. Інструмент нормалізації розпізнає такі марки на основі переліку відомих приватних брендів та коду країни; якщо назва ідентифікована як приватна марка, оркестратор завершує роботу зі статусом «private_label» без витрачання кроків на завідомо безрезультатний пошук.

Неоднозначні назви (наприклад, «Сіль 1 кг» або «Млієко» без зазначення бренду) потребують розширеної стратегії пошуку. У таких випадках інструмент нормалізації генерує більше пошукових запитів із різними уточненнями, а оркестратор виконує додаткові цикли пошуку для охоплення варіантів різних брендів. Інструмент оцінювання відповідності присвоює таким знахідкам нижчі оцінки впевненості, оскільки без конкретного бренду точна відповідність не може бути гарантована.

Розрізнення мультипаків та одиничних товарів (наприклад, «4×125g» проти «125g») забезпечується ключовими словами варіантів, які генерує інструмент нормалізації. Ці ключові слова зберігаються у робочому просторі (підпункт 3.4) та використовуються інструментами виокремлення та оцінювання для фільтрації знахідок за об'ємом або кількістю одиниць у пакуванні. Без цього механізму ціна мультипаку помилково порівнювалась би з ціною одиничного товару.

3.7 Впровадження модуля пошуку цін у застосунок Billka AI

Для впровадження модуля моніторингу цін у мобільний застосунок Billka AI розроблено прототип користувацького інтерфейсу у середовищі Figma [57]. Прототип визначає послідовність екранів, через яку користувач ініціює пошук цін та переглядає результати. Оскільки модуль працює у фоновому режимі та потребує декількох хвилин на обробку одного запиту, інтерфейс побудований за асинхронним принципом. Користувач ініціює пошук і отримує сповіщення після його завершення.

Сценарій використання складається з двох фаз: ініціювання пошуку та перегляд результатів. Послідовність екранів першої фази наведено на рисунку 3.5.

Користувач відкриває раніше збережений чек у застосунку (рис. 3.5 (а)). Чек відображається у вигляді переліку розпізнаних товарних позицій

із зазначенням ціни та кількості (рис. 3.5 (б)). Користувач обирає позицію, для якої хоче знайти альтернативні ціни, та у контекстному меню натискає кнопку відстеження ціни (рис. 3.5 (в)). Застосунок передає назву позиції, код країни, місто та назву магазину-джерела до модуля моніторингу цін (підпункт 3.5). На екрані з'являється повідомлення про початок пошуку (рис. 3.5 (г)): оскільки обробка одного запиту займає в середньому 3–4 хвилини, застосунок сповістить користувача після завершення.

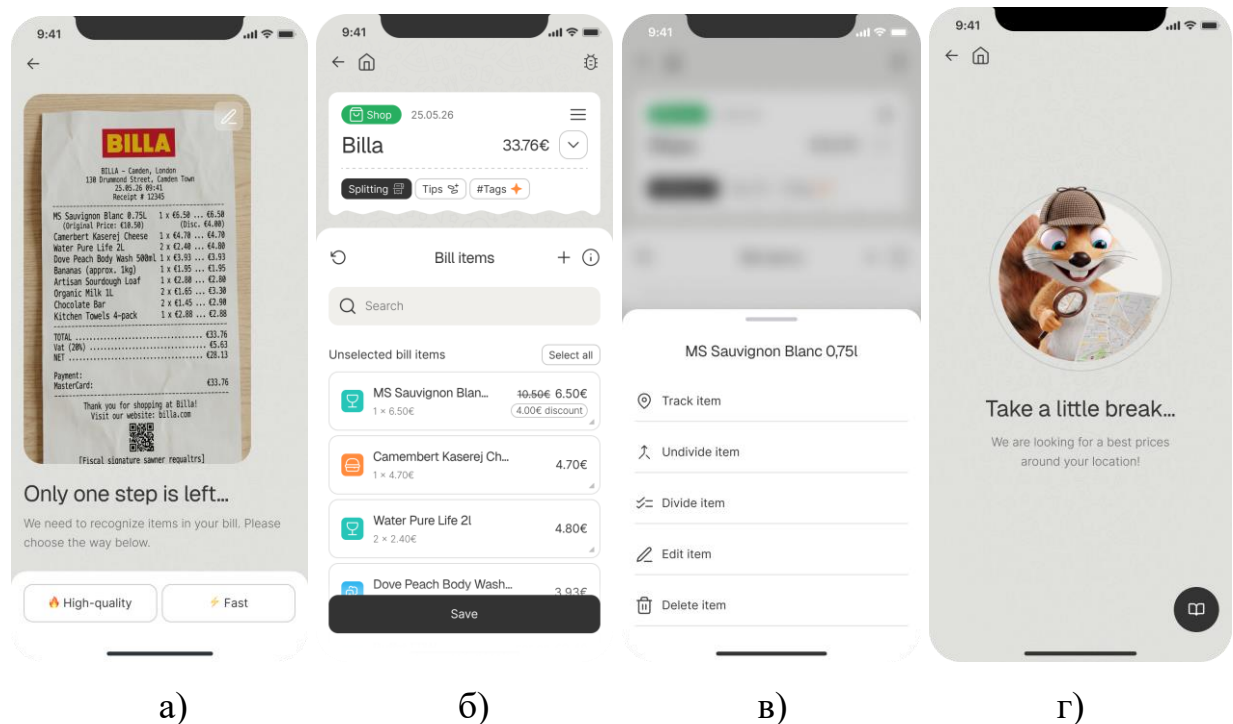


Рисунок 3.5 – Ініціювання пошуку цін:

а) фото чека; б) розпізнаний чек;

в) контекстне меню позиції; г) екран очікування

Після завершення пошуку користувач отримує сповіщення та переходить до перегляду результатів (рис. 3.6). На екрані відстежуваних товарів відображається короткий підсумок останнього пошуку для кожної позиції: кількість знайдених магазинів та діапазон цін (рис. 3.6 (а)). Користувач обирає товар та переглядає детальні результати (рис. 3.6 (б)). Знахідки згруповано за категорією відповідності на точні відповідності, схожі знахідки та інші варіанти. Для кожної знахідки зазначено назву

магазину, ціну та посилання на сторінку-джерело. Найдешевша точна відповідність виокремлена як рекомендована пропозиція. За наявності історії пошуків користувач може переглянути динаміку цін у часі (рис. 3.6 (в)) та розташування магазинів на карті (рис. 3.6 (г)).

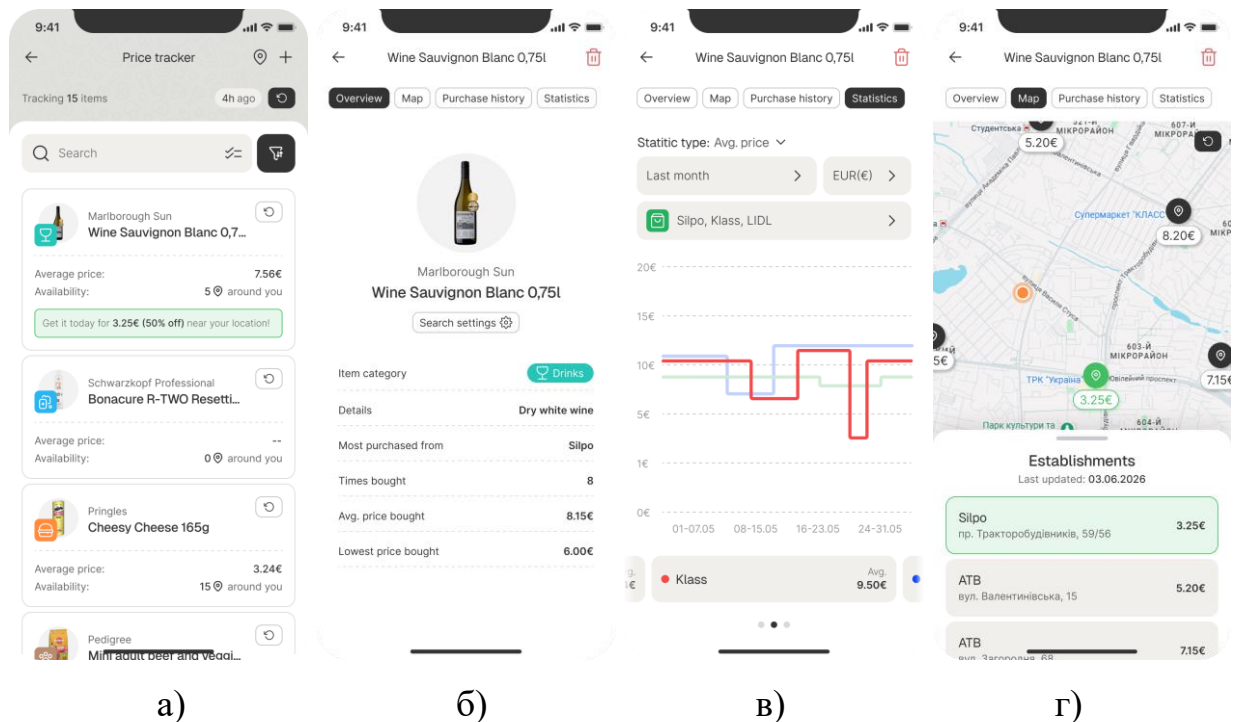


Рисунок 3.6 – Перегляд результатів пошуку:

- а) відстежувані товари; б) детальні результати;
в) історія цін; г) карта магазинів

Модуль повертає назви торговельних мереж, але не фізичні адреси окремих магазинів. Пошук найближчих точок мережі за геолокацією користувача та відображення їх на карті покладається на засоби застосунку Billka AI та не є функціональністю розробленого модуля.

3.8 Ілюстрація роботи модуля

Для демонстрації роботи модуля розглянуто обробку позиції «Mlieko Rajo Bezl 1,5 % 1l» із чека магазину Billa у місті Братислава (Словаччина). Ця

позиція обрана як представницька: назва містить аббревіатури («Bezl» – безлактозове, «Rajo» – бренд), що потребують нормалізації. Журнал виконання з міркуваннями оркестратора на кожному кроці зберігається у вихідному результаті модуля та доступний для діагностики. Діаграму трансформації даних під час обробки цієї позиції наведено на рисунку 3.7.

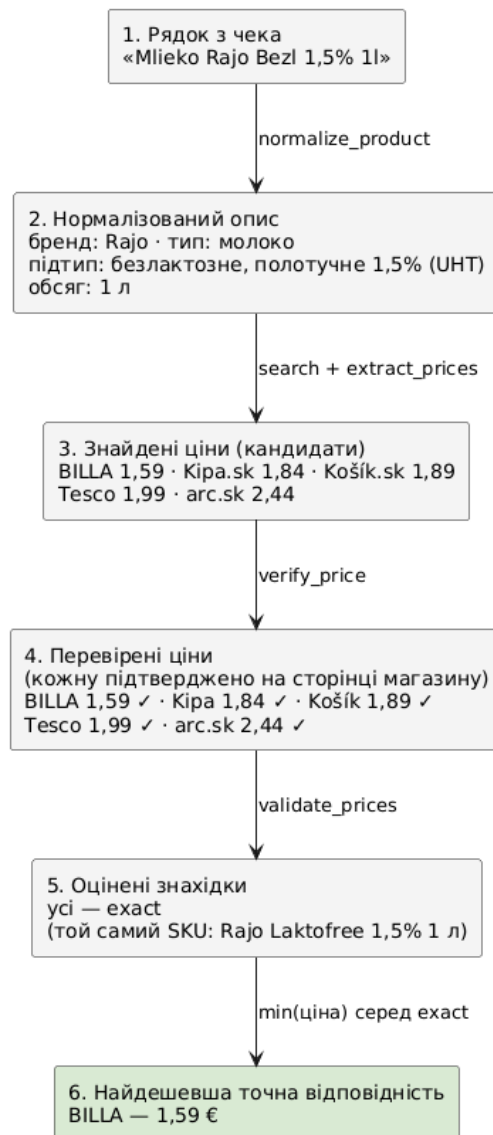


Рисунок 3.7 – Трансформація даних під час обробки позиції «Mlieko Rajo Bezl 1,5 % 1l»

Нормалізація запиту (крок 1). Інструмент нормалізації розібрав скорочену назву з чека та визначив структуровані поля: бренд Rajo, тип – trvanlivé mlieko (ультрапастеризоване молоко), підтип – bezlaktózové

(безлактозове), жирність 1,5 %, об'єм 1 л. Слід зазначити, що тип «trvanlivé» не вказано явно у назві з чека, тобто модель визначила його на основі знання продуктової лінійки бренду Ražo. На основі структурованих полів згенеровано 8 пошукових запитів різними мовами та формулюваннями, зокрема «Ražo mlieko bezlaktozove 1,5 % 1l cena», «Ražo Lakto Free milk price Bratislava» та «Ražo bezlaktozove mlieko polotučné cena». Різноманітність формулювань забезпечує покриття різних способів запису назви товару на сайтах магазинів.

Пошук торговельних точок (крок 2). Інструмент пошуку магазинів знайшов 47 торговельних мереж у радіусі 15 км від Братислави через OpenStreetMap. Серед яких є Billa, Tesco, Kaufland, Lidl, Košík.sk та інші. Цей перелік використовується оркестратором для формування цільових пошукових запитів та оцінки повноти охоплення.

Перший цикл пошуку та виокремлення (кроки 3–5). Оркестратор виконав два вебпошуки з різними формулюваннями та передав результати інструменту виокремлення цін. Перший запит є загальним пошуком за назвою товару, а другий за альтернативним написанням. Виокремлення виділило початковий перелік цінових пропозицій.

Другий цикл пошуку (кроки 6–9). Оркестратор оцінив охоплення магазинів як недостатнє та виконав ще три вебпошуки, спрямовані на конкретні магазини (Košík.sk, Tesco) та агрегатори цін. Друге виокремлення додало нові пропозиції до робочого простору.

Перша серія перевірок (кроки 10–14). Оркестратор обрав п'ять знахідок для перевірки на сторінках магазинів. Інструмент перевірки завантажив кожну сторінку через Firecrawl та підтвердив відповідність товару і ціни. Зокрема, на сторінці BILLA підтверджено ціну 1,59 € за товар «RAŽO LAKTOFREE MLIEKO 1,5 % UHT 1L», на сторінці Košík.sk – 1,89 €, на сторінці Tesco – 1,99 €.

Третій цикл пошуку та перевірки (кроки 15–19). Після першої серії перевірок оркестратор повернувся до пошуку для розширення охоплення. Він

виконав ще два вебпошуки з уточненими формулюваннями, виокремив додаткові пропозиції та перевірів ціни на двох нових сторінках. Цей адаптивний цикл є характерною особливістю агентного підходу, неможливою у фіксованій послідовності.

Оцінювання відповідності (крок 20). Інструмент оцінювання відповідності проаналізував усі накопичені пропозиції за чотирма критеріями: бренд, підтип, об'єм та додаткові атрибути. П'ять знахідок визнано точними відповідностями з високою оцінкою впевненості.

Формування відповіді (крок 21). Оркестратор завершив роботу та повернув структурований результат. Найдешевшою точною відповідністю визначено BILLA за ціною 1,59 €, що на 20 % дешевше за ціну покупки у тому самому магазині за іншою адресою (1,99 €). Повний перелік знахідок, відсортований за зростанням ціни, наведено у таблиці 3.1.

Таблиця 3.1 – Результати пошуку цін для позиції «Mlieko Ražo Bezl 1,5 % 1l»

Магазин	Ціна, €	Відповідність
BILLA	1,59	точна
Kipa.sk	1,84	точна
Košík.sk	1,89	точна
Tesco	1,99	точна
arc.sk	2,44	точна

Обробка зайняла 21 інструментний крок та тривала 167 секунд при вартості 0,122 долара.

Наведений приклад демонструє характерні особливості роботи модуля: збагачення скороченої назви з чека інформацією про продуктову лінійку бренду, багатомовний пошук із різними формулюваннями, три адаптивних цикли пошуку та перевірки (оркестратор двічі повертався до пошуку після перевірок для розширення охоплення магазинів), а також покрокова верифікація цін безпосередньо на сторінках магазинів.

3.9 Оцінка якості роботи модуля

Для оцінки якості розробленого модуля виконано незалежне тестування на наборі з 15 товарних позицій, які не входили до експериментального набору розділу 2. Набір сформовано з реальних чеків трьох країн по п'ять позицій на кожен. Це Україна (Харків, магазин АТБ), Словаччина (Братислава) та Австрія (Відень, Хайнбург-на-Дунаї). Товари обрано із наголосом на нові категорії та складні випадки, не представлені у наборі розділу 2: морепродукти, дріжджі, локальний сир, шпинат, банани (масовий товар без бренду), соус, шоколад із відсотком какао та смаком, вода (неповна назва), тунець (мультипак), ягоди (приватна марка), пиво (скорочена аббревіатура), консервована кукурудза.

Методика тестування ідентична застосованій у розділі 2. Модуль отримує лише назву позиції з чека, код країни та місто, а правильність кожної знахідки визначено через ручну розмітку за тими самими критеріями (підрозділ 2.1). Усього модуль повернув 128 знахідок, кожен з яких розмічено вручну. Результати за окремими позиціями наведено у таблиці 3.2.

Зведення результатів із порівнянням з даними розділу 2 та вимогами технічного завдання (підпункт 3.1) наведено у таблиці 3.3.

Усі п'ять вимог технічного завдання виконано. Точність на контрольному наборі (75 %) навіть перевищує результат розділу 2 (74 %) та строга точність зросла (63 % проти 60 %). Покриття магазинів суттєво зросло (6,2 проти 5,2), а вартість обробки знизилась (0,099 проти 0,114 долара). Модуль повернув знахідки для всіх 15 позицій без жодного порожнього результату. Ці дані свідчать, що модуль узагальнюється на нові товари та не є підігнаним під набір, використаний при виборі методу.

Найвищу точність показали позиції з відомим брендом та чіткою назвою: «Kukurica zlatá Bonduelle 212ml» (100 %), «MAN.Mozart Mignon» (100 %), «Mladý špenát 125g» (100 %) та «Заправка 220 г Торчин буряково-

томатна» (93 %). Найнижчу точність показали позиції з неповною назвою або приватною маркою, такі як «Vöslauer» (50 %) і «S-Budget Heidelbeere» (50 %).

Таблиця 3.2 – Результати тестування модуля на контрольному наборі

Позиція	Час, с	Вартість, \$	Кроків	Знайдено	Точність	Мін. ціна вірна
Креветки 0,7 кг Norven Vannamei	349	0,098	18	11	8/11 (73 %)	-
Сік 200 мл ОКЗДХ Наш Сік Яблучно- персик.	190	0,105	18	7	6/7 (86 %)	+
Дріжджі 22 г SAF- INSTANT м/уп	187	0,093	17	8	6/8 (75 %)	-
Морозиво 0,5 кг Magnat сорбет манго	219	0,098	19	10	6/10 (60 %)	-
Заправка 220 г Торчин буряково- томатна	223	0,105	18	14	13/14 (93 %)	+
Kukurica zlatá Bonduelle 212ml	165	0,072	14	5	5/5 (100 %)	+
Sabinovská hrudka	139	0,084	15	5	4/5 (80 %)	-
Mladý špenát 125g	115	0,050	13	6	6/6 (100 %)	+
Banány Bonvia	164	0,079	17	7	4/7 (57 %)	+
Lindt Creation 70% Orange 150g	266	0,131	21	10	6/10 (60 %)	-
MAN.Mozart Mignon	250	0,087	18	12	12/12 (100 %)	+
Vöslauer	260	0,136	20	8	4/8 (50 %)	-
Rio Mare Thunf. natur 3er	272	0,121	21	9	7/9 (78 %)	-
S-Budget Heidelbeere	292	0,127	21	6	3/6 (50 %)	-
ZW.DKL.BIER	341	0,092	17	10	6/10 (60 %)	+
Середнє	229	0,099	17,8	8,5	96/128 (75 %)	7/15 (47 %)

Таблиця 3.3 – Відповідність результатів вимогам технічного завдання

Вимога	Поріг	Результати на контрольній вибірці (розділ 3)	Результат на експериментальній вибірці (розділ 2)	Статус
Час на позицію	≤ 5 хв	3,8 хв (серед.)	3,4 хв	виконано
Точність	≥ 70 %	75 %	74 %	виконано
Охоплення магазинів	≥ 4 на товар	6,2	5,2	виконано
Повнота	≥ 90 %	100 % (15/15)	100 % (20/20)	виконано
Вартість на позицію	$\leq \$0,20$	\$0,099	\$0,114	виконано

Серед типових причин помилок на контрольному наборі можна виділити кілька категорій. Неповна назва на чеку, як-от «Vöslauer» без зазначення об'єму та типу води, через що у видачу потрапляють варіанти Sport та Flavoured. Підміна варіанту, наприклад, енергетик «S-Budget Energy Heidelbeere» замість замороженої чорниці «S-Budget Heidelbeere». Розбіжність бренду, коли сорбет «Рудь» потрапляє замість «Magnat». Розбіжність форми товару, наприклад, тунець «в олії» замість «natur» або креветки «варено-охолоджені» замість «варено-морожені».

ВИСНОВКИ

У рамках кваліфікаційної роботи розроблено інтелектуальний модуль для моніторингу цін товарів з паперового чеку та впроваджено його у мобільний застосунок для відстеження витрат Billka AI. Робота охоплює повний цикл від аналізу існуючих рішень до реалізації та оцінки якості модуля, та вирішено такі завдання:

- проведено аналіз існуючих сервісів порівняння цін та застосунків для відстеження витрат, що дало можливість визначити їхні обмеження та обґрунтувати потребу у модульному рішенні для інтелектуального зіставлення скорочених назв із касових чеків;

- проаналізовано підходи до вирішення задачі порівняння цін на основі вебскрейпінгу та LLM, що дало можливість визначити чотири концептуально різних підходи для подальшого експериментального порівняння;

- сформовано набір даних із 20 реальних позицій касових чеків трьох країн (Словаччина, Україна, Австрія) та виконано ручну розмітку усіх знайдених записів, що забезпечило об'єктивну базу для порівняння підходів;

- визначено критерії та метрики оцінки якості пошуку цін (точність, строга точність, охоплення магазинів, правильна найдешевша ціна, чесність самооцінки), що дало змогу кількісно порівняти підходи за єдиною шкалою;

- розроблено та реалізовано прототип підходу на основі вебскрейпінгу з використанням браузерів без графічного інтерфейсу та алгоритмів нечіткого порівняння рядків (точність 50 %, охоплення 1,6 магазину на товар);

- розроблено та реалізовано прототип підходу на основі послідовності LLM із залученням сервісів вебпошуку та завантаження сторінок (точність 57 %, охоплення 7,1 магазину на товар);

- розроблено та реалізовано прототипи двох агентних підходів за патерном ReAct: простий агент (точність 88 %, 5 позицій без знахідок) та гібридний агент зі спеціалізованими LLM-інструментами (точність 74 %, повне охоплення, найнижча вартість серед LLM-підходів);

– проведено порівняльний аналіз LLM різних розробників (OpenAI, Google, Anthropic) для кожного етапу роботи системи, що дало можливість обрати оптимальні моделі для оркестрації (gpt-5-mini) та спеціалізованих інструментів (gpt-4.1-mini, claude-haiku-4-5);

– проведено експериментальне порівняння чотирьох реалізованих підходів та за сукупністю критеріїв (надійність, вартість, точність, масштабованість) обрано підхід на основі гібридного агента для впровадження;

– сформовано підсумковий алгоритм роботи модуля, що складається з трьох етапів: отримання вхідних даних із застосунку, пошук цін гібридним агентом та подання результатів користувачеві;

– розроблено архітектуру модуля як Python-бібліотеки та впроваджено його у мобільний застосунок Billka AI.

Розглянуто існуючі сервіси порівняння цін та мобільні застосунки для відстеження витрат, проаналізовано літературні джерела, що стосуються задач автоматизованого моніторингу цін. Вивчено особливості формування назв товарів у касових чеках різних торговельних мереж та визначено, що задача порівняння цін на основі даних із чеків потребує інтелектуальної нормалізації скорочених назв, яка виходить за межі можливостей звичайного текстового пошуку.

Модуль реалізовано мовою Python із залученням моделей трьох провайдерів LLM та зовнішніх сервісів вебпошуку (Tavily), завантаження сторінок (Firecrawl) та пошуку торговельних точок (OpenStreetMap). Розроблено прототип користувацького інтерфейсу у середовищі Figma.

Незалежне тестування на контрольному наборі з 15 нових позицій підтвердило виконання всіх вимог технічного завдання: точність 75 %, охоплення 6,2 магазину на товар, повнота 100 %, вартість 0,099 долара на позицію, середній час обробки 3,8 хвилини.

Розглянуто перспективи покращення системи, які включають залучення візуальної інформації про товари на етапі нормалізації: користувач може сфотографувати придбані товари, що дасть модулю змогу

ідентифікувати продукт за зображенням упаковки, а не лише за скороченою назвою з чека; додатково можуть використовуватися фотографії товарів, знайдені під час пошуку в інтернеті, для підвищення точності зіставлення. У випадках, коли модуль не може однозначно визначити товар за назвою з чека, доцільно пропонувати користувачеві декілька варіантів ідентифікації на вибір. Серед інших напрямів покращення варто відзначити розширення набору даних для більш репрезентативної оцінки якості та впровадження механізму відстеження динаміки цін у часі.

Результати кваліфікаційної роботи можуть бути корисними для розробників мобільних застосунків обліку витрат, що прагнуть доповнити функціональність сканування чеків автоматичним порівнянням цін. Модульна архітектура бібліотеки дає змогу інтегрувати її у сторонні системи без розгортання окремого сервісу. Методика експериментального порівняння чотирьох підходів та сформований набір даних із ручною розміткою можуть слугувати відправною точкою для подальших досліджень у галузі інтелектуального зіставлення товарів із різних джерел.

Результати роботи апробовано у вигляді 2 тез доповідей під час Міжнародного молодіжного форуму «Радіoeлектроніка і молодь у XXI столітті» [58] та XI Міжнародної науково-практичної конференції «Globalization of scientific knowledge: international cooperation and integration of sciences» [59].

Практична цінність отриманих результатів підтверджується їх успішним впровадженням в ІТ-компанії ТОВ «Сайтосс». Розроблений інтелектуальний модуль інтегровано у програмне забезпечення для моніторингу витрат, що дозволило автоматизувати процес пошуку альтернативних цінових пропозицій на товари у вказаному радіусі на основі даних із касових чеків.

Дослідження виконано в межах проєкту INITIATE за грантом No.101136775 HORIZON WIDERA 2023 ACCESS 03.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Гороховатський В., Творошенко І. (2026) Продуктивні моделі аналізу даних у методах розпізнавання зображень: монографія. Харків: ХНУРЕ, 153 с.
2. Cherednichenko O., Yakovleva O., and Hrechyshkin D. (2025). *A data-centric approach to crowd counting: Synthetic dataset creation and evaluation*. In S. Vladov, Y. Bodyanskiy, V. Lytvyn, I. Aizenberg, & L. Chyrun (Eds.), *Proceedings of the International Workshop on Applied Intelligent Security Systems in Law Enforcement (AISSLE-2025)* CEUR Workshop Proceedings. pp. 22–34.
3. Yakovleva O., Matúšová S., Tvoroshenko I., Isaiev Y. (2024). Visitor counting based on video stream analysis from surveillance cameras. *Scientific Journal of Bratislava University of Economics and Management «Public Administration and Regional Development, Economics, Management and Marketing»*, vol. 20, no. 1, pp. 67–87.
4. Yakovleva O., Kovtunenکو A., Liubchenko V., Honcharenko V., and Kobylin O. (2023). Face Detection for Video Surveillance-based Security System. *CEUR Workshop Proceedings Vol. 3403*. pp. 69-86.
5. Yakovleva O., Kovač M., Ardasov V., and Yeremenko I. (2023). Study on adding functionality to the Zoom online conference system for monitoring the participant activities. *Public Administration and Regional Development*, 19(1), pp. 158–183.
6. Yakovleva O., Nebeský L, Liakhov P. (2023). Research methods of texture image analysis to solve the texture search problem. *Proceedings of the IV International Scientific and Practical Conference «The world of modern technologies and inventions»*. Vienna, Austria. 2023. pp. 252-261.
7. Ковтуненко А.Р., Яковлева О.В., Любченко В.А., Янголенко О.В. (2020). Дослідження сумісного використання математичної морфології та згорткових нейронних мереж для вирішення задачі розпізнавання цінників. *Вісник Національного технічного університету ХПІ* (3). 24-31.

8. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2026) Feature space quantization and application of metrics in structural methods of image recognition, *IEEE Access*, vol. 14, pp. 17812-17824.
9. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2025) Image description compression in classification structural methods, *IEEE Access*, vol. 13, pp. 43631-43641.
10. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, 33(1), pp. 113-125.
11. Gorokhovatskyi O., Yakovleva O. (2024). Medoids as a packing of ORB image descriptors. *Advanced Information Systems*, 8(2), 5–11.
12. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.
13. Yakovleva O., Nikolaieva K. (2020). Research Of Descriptor Based Image Normalization And Comparative Analysis Of SURF, SIFT, BRISK, ORB, KAZE, AKAZE Descriptors. *Advanced Information Systems*, 4(4), 89-101.
14. Yakovleva O., Matúšová S., Liubchenko V., Maksimov H. (2025). Innovative solutions based on AI in education, on the example of generating multimodal lecture notes. *Scientific Journal of Bratislava University of Economics and Management «Public Administration and Regional Development, Economics, Management and Marketing»*, vol. 21, no. 1, pp. 140-163.
15. Остапчук В.О., Яковлева О.В. (2026) Порівняльний аналіз OCR та LLM підходів для перекладу тексту з зображення сторінки манги. 30-й Міжнародний молодіжний форум «Радіoeлектроніка і молодь у XXI столітті». Збірник матеріалів форуму, 22–24 квітня 2026 року, Харків, Україна, Т. 7, С. 127-129.

16. Datsok Y., Yakovleva O. (2026) Automated identification of medicinal products in mobile information systems based on multimodal models. Інформаційні технології: наука, техніка, технологія, освіта, здоров'я. Тези доповідей XXXIV міжнародної науково-практичної конференції MicroCAD-2026 (13–16 травня 2026 року). Харків: НТУ «ХПІ», С. 1692.

17. Yakovleva O., Sokolova A., Datsok Y. (2026) Optimization of multimodal Gemini models for food product composition recognition and explanation with automated translation into Ukrainian. Theoretical and empirical scientific research: concept and trends: proceedings of the X international scientific and practical conference, May 8, 2026, Oxford, United Kingdom, pp. 117-120.

18. Köpcke H., and Rahm E. (2010) Frameworks for entity matching: A comparison, Data & Knowledge Engineering, 69(2), pp. 197-210.

19. Google Shopping. URL: <https://shopping.google.com> (дата звернення 25.04.2026).

20. About idealo. URL: <https://www.idealco.uk/company/about-us/about-idealco> (дата звернення 25.04.2026).

21. About PriceRunner. URL: <https://www.pricerunner.com/info/about-pricerunner> (дата звернення 25.04.2026).

22. Hotline.ua. URL: <https://apps.apple.com/ua/app/hotline-ua/id1080498627> (дата звернення 25.04.2026).

23. Tabletki.ua. URL: <https://tabletki.ua> (дата звернення 25.04.2026).

24. Apteki.ua. URL: <https://apteki.ua> (дата звернення 25.04.2026).

25. Liki.ua. URL: <https://liki.ua> (дата звернення 25.04.2026).

26. Де Дешевше. URL: <https://apps.apple.com/ua/app/де-дешевше/id6738097657> (дата звернення 25.04.2026).

27. GoToShop.ua. URL: <https://play.google.com/store/apps/details?id=ua.gotoshop> (дата звернення 25.04.2026).

28. Kassalapp. URL: <https://apps.apple.com/no/app/kassalapp/id1545547638> (дата звернення 25.04.2026).

29. Enhver. URL: <https://enhver.no> (дата звернення 25.04.2026).
30. Heureka.sk. URL: <https://www.heureka.sk> (дата звернення 25.04.2026).
31. Priceo.sk. URL: <https://priceo.sk> (дата звернення 25.04.2026).
32. ChangeDetection.io. URL: <https://changedetection.io> (дата звернення 25.04.2026).
33. Billka AI. URL: <https://billka.sytoss.com/en/> (дата звернення 25.04.2026).
34. SyToss – компанія-розробник застосунку Billka AI. URL: <https://www.sytoss.com/billka-ai> (дата звернення 25.04.2026).
35. Splitwise. URL: <https://www.splitwise.com> (дата звернення 25.04.2026).
36. Smart Receipts. URL: <https://www.smartreceipts.co> (дата звернення 25.04.2026).
37. Playwright. URL: <https://playwright.dev> (дата звернення 25.04.2026).
38. Patchright. URL: <https://github.com/Kaliiiiiiiiii-Vinyzu/patchright> (дата звернення 25.04.2026).
39. Beautiful Soup. URL: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> (дата звернення 25.04.2026).
40. RapidFuzz. URL: <https://github.com/maxbachmann/RapidFuzz> (дата звернення 25.04.2026).
41. Levenshtein V.I. (1966) Binary codes capable of correcting deletions, insertions, and reversals, Soviet Physics Doklady, 10(8), pp. 707-710.
42. Introducing ChatGPT. URL: <https://openai.com/index/chatgpt/> (дата звернення 25.04.2026).
43. Chatbot Arena: Open platform for crowdsourced AI benchmarking. URL: <https://lmarena.ai> (дата звернення 25.04.2026).
44. Yao S., Zhao J., Yu D., Du N., Shafran I., Narasimhan K., and Cao Y. (2023) ReAct: Synergizing reasoning and acting in language models. URL: <https://arxiv.org/abs/2210.03629> (дата звернення 25.04.2026).

45. Building effective agents. URL: <https://www.anthropic.com/engineering/building-effective-agents> (дата звернення 25.04.2026).
46. OpenStreetMap. URL: <https://www.openstreetmap.org> (дата звернення 25.04.2026).
47. Overpass API. URL: <https://overpass-api.de> (дата звернення 25.04.2026).
48. Suprun A., Tvoroshenko I., Gorokhovatskyi V., and Yakovleva O. (2025) Development and research of a method for the combined use of large language models for text generation, *International Journal of Academic and Applied Research*, 9(10), pp. 249-263.
49. Dice L.R. (1945) Measures of the amount of ecologic association between species, *Ecology*, 26(3), pp. 297-302.
50. Jaccard P. (1912) The distribution of the flora in the alpine zone, *New Phytologist*, 11(2), pp. 37-50.
51. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Advanced Information Systems*, 7(1), С. 5-13.
52. Bohdan N., Tvoroshenko I., Gorokhovatskyi V., and Kobylin O. (2025) Development of a hybrid method to enhance context memory for a chatbot application based on large language models, *International Journal of Academic Information Systems Research*, 9(10), pp. 7-18.
53. Tavily – AI-native search and extraction API. URL: <https://tavily.com> (дата звернення 21.05.2026).
54. Firecrawl – turn websites into LLM-ready data. URL: <https://firecrawl.dev> (дата звернення 21.05.2026).
55. Pydantic – Data validation using Python type annotations. URL: <https://docs.pydantic.dev> (дата звернення 01.06.2026).

56. Nii H.P. (1986) Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures, AI Magazine, 7(2), pp. 38–53.

57. Figma – Collaborative interface design tool. URL: <https://www.figma.com> (дата звернення 01.06.2026).

58. Белевець Р.В., Яковлева О.В. (2026) Порівняння підходів на основі парсингу та ШІ агентів для вирішення задачі відстеження цін товарів. 30-й Міжнародний молодіжний форум «Радіоелектроніка і молодь у XXI столітті». Збірник матеріалів форуму, 22–24 квітня 2026 року, Харків, Україна, Т. 7, С. 32-34.

59. Яковлева О., Белевець Р. (2026) Архітектура автономного ШІ-агента для порівняння цін товарів з касових чеків. Globalization of scientific knowledge: international cooperation and integration of sciences: proceedings of the XI International Scientific and Practical Conference, May 15, 2026, Vinnytsia, Ukraine – Vienna, Austria (Online). Грааль науки, № 68, С. 1077-1080.