

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ФАХОВИЙ БІЗНЕС-КОЛЕДЖ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
(повна назва випускної кафедри)

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему:

**СИСТЕМА РЕКОМЕНДАЦІЙ ДЛЯ ОНЛАЙН-ПЛАТФОРМИ З
ПРОДАЖУ ПРОДУКТІВ ХАРЧУВАННЯ**

Виконав:

студент групи 2КІ-23 зі спеціальності

123 – «Комп'ютерна інженерія»

Олександр БОРИСЕНКО

Науковий керівник:

к.т.н Роксолана БРЕУС

Черкаси, 2025

АНОТАЦІЯ

Дипломна робота присвячена комплексному аналізу сучасних рекомендаційних систем у сфері онлайн-торгівлі продуктами харчування та розробці проекту власної рекомендаційної системи, яка враховує особливості електронної комерції. У роботі детально розглянуто поточний стан ринку онлайн-продажів продуктів харчування, а також роль рекомендаційних систем у підвищенні рівня задоволеності користувачів та збільшенні продажів. Проведено огляд і аналіз найбільш відомих платформ (Amazon Fresh, Instacart, Rozetka тощо), їхніх алгоритмів та функціональних можливостей, з визначенням сильних та слабких сторін існуючих рішень.

Окрім цього, визначено функціональні та нефункціональні вимоги до майбутньої системи, включаючи вимоги до продуктивності, масштабованості, зручності користувацького інтерфейсу, а також проблеми безпеки і захисту приватності персональних даних. На основі зібраних даних розроблено архітектуру системи, описано основні компоненти та сценарії її використання. Практична частина роботи включає проектування структури рекомендаційної системи, що сприятиме ефективній обробці та зберіганню даних для генерації персоналізованих рекомендацій.

Результати дослідження можуть бути використані для підвищення конкурентоспроможності онлайн-магазинів продуктів харчування, а також для покращення користувацького досвіду шляхом надання точних і релевантних рекомендацій. Робота має практичну цінність для розробників та бізнес-аналітиків у сфері електронної комерції.

Ключові слова: РЕКОМЕНДАЦІЙНА СИСТЕМА, ОНЛАЙН-ТОРГІВЛЯ, ПРОДУКТИ ХАРЧУВАННЯ, ЕЛЕКТРОННА КОМЕРЦІЯ, АЛГОРИТМИ РЕКОМЕНДАЦІЙ, ПЕРСОНАЛІЗАЦІЯ, БЕЗПЕКА ДАНИХ.

ABSTRACT

The thesis is devoted to a comprehensive analysis of modern recommender systems in the field of online food trading and the development of a project of its own recommender system that takes into account the peculiarities of e-commerce. The paper analyzes in detail the current state of the online food sales market, as well as the role of recommender systems in increasing user satisfaction and increasing sales. The author reviews and analyzes the most well-known platforms (Amazon Fresh, Instacart, Rozetka, etc.), their algorithms and functionalities, and identifies the strengths and weaknesses of existing solutions.

In addition, we identified functional and non-functional requirements for the future system, including performance, scalability, user interface, security, and privacy issues. Based on the collected data, the system architecture is developed, the main components and scenarios of its use are described. The practical part of the work includes designing the structure of the recommendation system, which will facilitate efficient data processing and storage for generating personalized recommendations.

The results of the study can be used to increase the competitiveness of online food stores, as well as to improve the user experience by providing accurate and relevant recommendations. The work has practical value for developers and business analysts in the field of e-commerce.

Keywords: RECOMMENDER SYSTEM, ONLINE TRADING, FOOD, E-COMMERCE, RECOMMENDATION ALGORITHMS, PERSONALISATION, DATA SECURITY.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. ОСОБЛИВОСТІ СИСТЕМИ ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ	5
1.1. Сучасний стан онлайн-торгівлі продуктами харчування	5
1.2. Сутність рекомендаційних систем та їх роль в електронній комерції	16
1.3. Основні алгоритми сучасних рекомендаційних систем	21
РОЗДІЛ 2. АНАЛІЗ ІСНУЮЧИХ РЕКОМЕНДАЦІЙНИХ СИСТЕМ У СФЕРІ ОНЛАЙН-ПРОДАЖУ ПРОДУКТІВ	24
2.1. Огляд популярних сервісів (Amazon Fresh, Instacart, Rozetka та ін.)	24
2.2. Аналіз сильних і слабких сторін реалізованих рішень	27
2.3. Порівняння підходів до рекомендацій	29
РОЗДІЛ 3. ФУНКЦІОНАЛЬНІ ТА НЕФУНКЦІОНАЛЬНІ ВИМОГИ ДО МАЙБУТНЬОЇ СИСТЕМИ	32
3.1. Визначення основних функцій системи	32
3.2. Вимоги до продуктивності, масштабованості та UX	33
3.3. Проблеми безпеки та приватності користувацьких даних	36
РОЗДІЛ 4. ПРОЄКТУВАННЯ СТРУКТУРИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ	40
4.1. Сценарії використання (Use Case діаграми)	40
4.2. Схема взаємодії компонентів (Data Flow Diagram)	46
4.3. Попередня модель зберігання та обробки даних	56
ВИСНОВКИ	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	61
ДОДАТОК А	
ДОДАТОК Б	
ДОДАТОК В	

ВСТУП

У сучасному цифровому світі онлайн-торгівля продуктами харчування стрімко розвивається, стаючи невід'ємною частиною повсякденного життя споживачів. Зростання попиту на швидкі, зручні та персоналізовані сервіси змушує компанії постійно вдосконалювати свій функціонал. Одним із ключових напрямів такого вдосконалення є впровадження рекомендаційних систем, що забезпечують користувачам індивідуальний підхід і сприяють зростанню продажів.

Актуальність теми зумовлена потребою в інтелектуальних інструментах, які здатні адаптуватися до індивідуальних уподобань клієнтів, аналізувати великі обсяги даних і формувати релевантні рекомендації у сфері онлайн-торгівлі продуктами. Такий підхід дозволяє не лише підвищити лояльність користувачів, але й оптимізувати бізнес-процеси підприємств електронної комерції.

Метою кваліфікаційної роботи є розробка концепції ефективної рекомендаційної системи для онлайн-платформи з продажу продуктів харчування з урахуванням специфіки поведінки користувачів та вимог до сучасних інформаційних сервісів.

Для досягнення цієї мети поставлено такі завдання:

- проаналізувати стан онлайн-торгівлі продуктами харчування в Україні та світі;
- дослідити теоретичні основи побудови рекомендаційних систем;
- охарактеризувати основні алгоритми рекомендаційних систем та визначити їх застосовність у даній предметній області;
- здійснити порівняльний аналіз існуючих платформ з рекомендаційним функціоналом;
- сформулювати функціональні та нефункціональні вимоги до майбутньої системи;

- запропонувати структуру та логіку проектування системи рекомендацій.

Об'єктом дослідження є система рекомендацій для онлайн-платформи продажу продуктів харчування.

Предметом дослідження виступають моделі, методи та інструменти побудови рекомендаційних систем для персоналізованих пропозицій у сфері E-Commerce.

Методи дослідження, що використовуються в роботі, включають: аналіз літературних джерел, системний аналіз існуючих платформ, методи класифікації та сегментації користувачів, моделювання архітектури IT-систем.

Практичне значення роботи полягає у створенні концептуальної моделі системи, яка може бути використана для впровадження персоналізованих рекомендацій у діючих або нових платформах з онлайн-торгівлі продуктами харчування.

Структура роботи включає вступ, чотири розділи, висновки, список використаних джерел і додатки. У першому розділі здійснено теоретико-аналітичний огляд сучасних рекомендаційних систем. У другому – представлено аналіз реальних прикладів реалізації таких систем у сфері онлайн-торгівлі продуктами. У третьому – визначено вимоги до майбутньої системи. У четвертому – запропоновано структуру рекомендаційної системи та описано її проектну модель.

РОЗДІЛ 1 ОСОБЛИВОСТІ СИСТЕМИ ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ

1.1 Сучасний стан онлайн-торгівлі продуктами харчування

Онлайн-торгівля продуктами харчування переживає період стрімкого зростання та трансформації, що особливо прискорилося після пандемії COVID-19. Цей сегмент електронної комерції характеризується унікальними особливостями та викликами, які відрізняють його від інших галузей онлайн-торгівлі [1].

Світовий ринок онлайн-продажу продуктів харчування демонструє вражаючі темпи зростання. За даними аналітичних компаній, обсяг ринку grocery E-Commerce у 2023 році перевищив 400 мільярдів доларів США, а прогнозовані темпи щорічного зростання (CAGR) складають близько 15-20% до 2030 року. Найбільший розвиток спостерігається у США, Китаї, країнах Європейського Союзу та Великій Британії.

Основними драйверами зростання є зміна споживчих звичок, особливо серед молодого покоління, підвищення рівня цифрової грамотності населення, розвиток логістичної інфраструктури та впровадження інноваційних технологій доставки. Пандемія COVID-19 стала каталізатором, який прискорив прийняття онлайн-шопінгу продуктів харчування навіть серед традиційно консервативних споживачів.

Онлайн-торгівля продуктами харчування має ряд специфічних характеристик, які відрізняють її від інших сегментів електронної комерції. По-перше, це необхідність забезпечення належних умов зберігання та транспортування, особливо для швидкопсувних товарів. По-друге, високий рівень довіри споживачів до якості та свіжості продуктів, що вимагає прозорості у відображенні інформації про товари [2].

Крім того, для цього сегменту характерна висока частота повторних покупок, оскільки продукти харчування є товарами щоденного вжитку. Це

створює унікальні можливості для побудови довгострокових відносин з клієнтами та застосування персоналізованих підходів до обслуговування.

Ландшафт онлайн-торгівлі продуктами харчування формують різноманітні типи компаній. Amazon через свої сервіси Amazon Fresh та Whole Foods Market займає лідируючі позиції у США. У Європі значну роль відіграють такі компанії як Ocado (Великобританія), Picnic (Нідерланди), та національні мережі супермаркетів, які активно розвивають онлайн-канали.

В Україні ринок представлений такими гравцями як Zakaz.ua, Glovo, Rozetka, а також власними онлайн-платформами великих ритейлерів - АТБ, Сільпо, Novus. Кожен з цих гравців використовує різні бізнес-моделі: від маркетплейсів до власних складських мереж та партнерських програм з локальними магазинами [3].

Сучасні онлайн-платформи продажу продуктів харчування активно впроваджують передові технології для покращення користувацького досвіду. Штучний інтелект та машинне навчання використовуються для персоналізації рекомендацій, прогнозування попиту, оптимізації логістичних маршрутів та управління запасами.

Технології доповненої реальності (AR) дозволяють споживачам краще оцінити продукти перед покупкою, а голосові асистенти спрощують процес замовлення. Блокчейн технології впроваджуються для забезпечення прозорості ланцюгів поставок та підтвердження походження органічних продуктів.

Незважаючи на позитивну динаміку розвитку, онлайн-торгівля продуктами харчування стикається з рядом серйозних викликів. Основними з них є високі логістичні витрати, особливо для доставки холодильних товарів, складність управління запасами швидкопсувних продуктів, та необхідність забезпечення високої якості обслуговування клієнтів [4].

Також значною проблемою є відносно низька маржинальність багатьох продуктів харчування, що ускладнює досягнення прибутковості онлайн-

операцій. Багато компаній досі перебувають у пошуку оптимальної бізнес-моделі, яка б дозволила забезпечити сталий прибуток при збереженні конкурентоспроможних цін.

В умовах зростаючої конкуренції та необхідності підвищення ефективності онлайн-платформ, системи рекомендацій стають критично важливим інструментом. Вони дозволяють не лише покращити користувацький досвід через персоналізацію, але й вирішувати бізнес-задачі: збільшення середнього чека, підвищення частоти покупок, зменшення відсотка відмов від кошика.

Для сегменту продуктів харчування системи рекомендацій мають особливе значення через специфіку споживання: регулярність покупок, сезонність попиту, дієтичні обмеження та персональні преференції. Ефективна система рекомендацій може суттєво спростити процес вибору для споживача, особливо враховуючи широкий асортимент сучасних онлайн-платформ.

Майбутнє онлайн-торгівлі продуктами харчування пов'язане з подальшою інтеграцією технологій, розвитком омніканальних підходів та підвищенням рівня персоналізації. Очікується зростання ролі автоматизованих рішень, включаючи автономні транспортні засоби для доставки та роботизовані склади.

Також прогнозується розвиток концепції "smart kitchen" та інтеграція онлайн-платформ з розумними домашніми приладами, що дозволить автоматизувати процес планування покупок та поповнення запасів. Системи рекомендацій у цьому контексті будуть відігравати ключову роль у забезпеченні плавної та інтуїтивної взаємодії між різними компонентами екосистеми [5].

Сучасний розвиток онлайн-торгівлі продуктами харчування характеризується кількома ключовими тенденціями, які формують майбутнє цієї індустрії та визначають стратегічні напрямки розвитку компаній у цій сфері.

Глобальний ринок онлайн-продажу продуктів харчування демонструє вражаючі темпи зростання. Обсяг світового ринку досяг 57,91 мільярда доларів США у 2024 році, а прогнозується його зростання до 623,70 мільярда доларів до 2034 року. За різними оцінками аналітичних компаній, очікуваний середньорічний темп зростання (CAGR) складає 26,83% протягом прогнозованого періоду 2025-2034 років.

Альтернативні прогнози також демонструють оптимістичні очікування: деякі дослідження передбачають досягнення обсягу ринку 1435,51 мільярда доларів до 2029 року з темпом зростання 21,5%. Ці цифри підтверджують, що онлайн-торгівля продуктами харчування є одним з найбільш динамічних сегментів електронної комерції [6].

Штучний інтелект стає центральним елементом трансформації галузі онлайн-продажу продуктів харчування. Алгоритми ШІ оптимізують маршрути доставки, передбачають час доставки та навіть прогнозують умови дорожнього руху, забезпечуючи швидшу, ефективнішу та економічно вигідну доставку.

Розумні системи управління запасами, автоматизовані служби доставки та оптимізовані методи ведення сільського господарства працюють на основі ШІ, роблячи продовольчу систему більш ефективною та сталою. Ця технологічна еволюція обіцяє покращити не лише процес приготування їжі, але й способи виробництва та доставки продуктів споживачам.

Персоналізація стає критично важливим фактором успіху в онлайн-торгівлі продуктами харчування. Компанії використовують дані про клієнтів для надання персоналізованих меню, цільових промо-акцій та індивідуальних графіків доставки на основі переваг клієнтів та історії замовлень.

ШІ допомагає постачальникам продовольчих послуг персоналізувати взаємодію з клієнтами, аналізуючи минулу поведінку та переваги, що дозволяє надавати індивідуальні рекомендації меню та персональні промо-пропозиції,

підвищуючи лояльність клієнтів. Такий підхід значно покращує задоволеність споживачів, змушуючи їх відчувати себе цінними та зрозумілими.

Екологічна свідомість споживачів формує новий тренд у галузі онлайн-доставки продуктів харчування. Сталий розвиток більше не є додатковою перевагою - це необхідність. У 2025 році користувачі схильються до брендів з екологічно свідомим позиціонуванням.

Очікується поява практик екологічної доставки, включаючи використання перероблених пакувальних матеріалів та доставку електричними транспортними засобами для зменшення забруднення атмосфери. Компанії активно впроваджують вуглецево-нейтральні варіанти доставки та інші екологічні рішення для зменшення впливу на навколишнє середовище [7].

Значним трендом стає зростання популярності приватних торгових марок. Згідно з дослідженням NIQ Mid-Year Consumer Outlook, 50% глобальних респондентів повідомляють про збільшення покупок продуктів приватних торгових марок. Це означає, що бренди продуктів харчування повинні докладати зусиль для конкуренції з зростанням приватних марок, включаючи розбудову власної ідентичності та унікальних пропозицій.

Структура онлайн-продажів продуктів харчування також зазнає змін. Основні продукти та кулінарні інгредієнти лідирують на ринку з приблизно 28,9% частки ринку в 2024 році, представляючи найбільший сегмент. Це свідчить про те, що споживачі все частіше використовують онлайн-платформи для закупівлі базових продуктів для домашнього приготування їжі.

Різні регіони демонструють унікальні патерни розвитку. У США онлайн-продажі продуктів харчування досягли 9,8 мільярда доларів у квітні 2025 року, що становить зростання на 15,2% порівняно з квітнем 2024 року, коли вони склали 8,5 мільярда доларів. Починаючи з серпня 2024 року, щомісячні онлайн-продажі продуктів харчування у США перевищують 9,5 мільярда доларів.

Споживчі очікування продовжують еволюціонувати у напрямку більшої зручності та якості обслуговування. У 2025 році клієнти очікують персоналізованих меню, кураторських рекомендацій та індивідуальних пропозицій. Використання AI-аналітики дозволяє платформам доставки їжі робити кожне замовлення унікальним для конкретного користувача [8].

Майбутнє галузі тісно пов'язане з впровадженням інноваційних логістичних технологій. Доставка їжі стає значно розумнішою завдяки прогнозуючому замовленню, автономним кухням та роботам-кур'єрам, роблячи страви більш доступними, швидшими та гіперперсоналізованими.

Ці глобальні тренди формують фундамент для розуміння сучасного стану та перспектив розвитку онлайн-торгівлі продуктами харчування, створюючи контекст для аналізу ролі та значення систем рекомендацій у цій динамічній індустрії.

Розвиток онлайн-торгівлі продуктами харчування характеризується суттєвими регіональними відмінностями, що зумовлені культурними особливостями, рівнем технологічного розвитку, логістичною інфраструктурою та споживчими звичками населення різних регіонів світу, зокрема:

- Азіатско-Тихоокеанський регіон – лідер глобального зростання. Азіатско-Тихоокеанський регіон домінує на ринку онлайн-торгівлі продуктами харчування у 2025 році. Обсяг ринку онлайн-продажу продуктів харчування в регіоні склав 139,61 мільярда доларів США у 2022 році з очікуваним щорічним темпом зростання 24,01% [9].

- Китай – найбільший ринок у світі. Китай займає провідне місце в глобальному ринку онлайн-торгівлі продуктами харчування. Основними складовими є:

- а) Alibaba Group – через свої платформи Taobao та Tmall надає широкий спектр продуктів харчування з розвиненою системою логістики та швидкої доставки. Компанія впровадила інноваційні концепції, такі як «Freshippo»

(Нема) – гібридні супермаркети, що поєднують офлайн та онлайн досвід покупок.

b) JD.com – відомий своєю власною логістичною мережею та гарантією якості продуктів. Платформа спеціалізується на швидкій доставці свіжих продуктів протягом години в мегаполісах.

c) Meituan – лідер у сфері доставки готової їжі та продуктів харчування, який обслуговує понад 600 мільйонів активних користувачів щомісячно. Фрукти та овочі були провідними продуктами, що купувалися онлайн у Китаї та Австралії, що демонструє специфіку споживчих переваг у регіоні.

- Індійський ринок онлайн-продажу продуктів харчування демонструє експоненціальне зростання завдяки цифровізації та зміні споживчих звичок. Серед ключових гравців на індійському ринку онлайн-торгівлі продуктами харчування виступають:

a) BigBasket - найбільша платформа онлайн-продажу продуктів в Індії, що пропонує понад 40,000 найменувань товарів і здійснює доставку в більш ніж 300 міст.

b) Zepto - стартап, що спеціалізується на ультрашвидкій доставці (10-15 хвилин) у великих містах, використовуючи мережу мікро-складів.

c) Blinkit (раніше Grofers) - фокусується на доставці продуктів протягом години з акцентом на свіжі та органічні продукти.

- Південна Корея та Японія має технологічні інновації. У Південній Кореї провідними онлайн-продуктами були заморожені продукти, що відображає особливості споживчих переваг та високий рівень технологічного розвитку системи доставки холодильних товарів.

- Європа – зрілий ринок з фокусом на якість. Європейський ринок онлайн-торгівлі продуктами харчування очікується досягти 797,34 мільярда доларів США у 2033 році порівняно з 178,51 мільярда доларів у 2024 році з темпом зростання 18,09% протягом 2025-2033 років. Продажі продуктів

харчування в Європі зросли на 2,4% у 2024 році, що трохи перевищує рівень інфляції на продукти харчування в 2,3%.

Великобританія – новинка європейського ринку. Їх основні представники [10]:

а) Ocado – британська компанія, що стала глобальним лідером у сфері автоматизованих складських технологій для онлайн-торгівлі продуктами харчування. Компанія розробила унікальні роботизовані склади, які стали еталоном для індустрії.

б) Tesco – найбільший британський ритейлер успішно інтегрував онлайн та офлайн канали, пропонуючи послуги click-and-collect та доставки додому.

Німеччина та Нідерланди мають фокус на сталість:

а) Picnic (Нідерланди) – інноваційна компанія, що використовує електричні фургони для доставки та застосовує алгоритми машинного навчання для оптимізації маршрутів і прогнозування попиту.

б) REWE (Німеччина) - один з провідних німецьких ритейлерів активно розвиває онлайн-сервіси з акцентом на органічні та локальні продукти.

Франція обрала розвиток унікальності. Carrefour впровадив концепцію click-and-collect у сотнях локацій, дозволяючи клієнтам замовляти онлайн та забирати покупки в зручний час.

- Північна Америка обирає технологічний прогрес та масштабність, адже у США найбільший західний ринок. Продажі онлайн-продуктів харчування різко зросли у другій половині 2024 року, збільшившись на 17,7% порівняно з попереднім роком:

а) Amazon – через Amazon Fresh та придбання Whole Foods Market створив екосистему, що поєднує онлайн-платформу, фізичні магазини та інноваційні технології доставки, включаючи Amazon Go (магазини без кас).

б) Walmart – найбільший ритейлер США активно конкурує з Amazon через сервіси Walmart+ та розвиток власної логістичної мережі для доставки продуктів.

с) Instacart – лідер у сфері персональних покупок та доставки, що співпрацює з тисячами локальних та національних ритейлерів.

- Канада адаптує американську модель. Loblaws – провідний канадський ритейлер розвиває онлайн-платформу PC Express з послугами click-and-collect та доставки додому.

- Латинська Америка перебуває у початковій стадії розвитку.

- Бразилія та Мексика є основними ринками регіону:

- a) Rappi – колумбійський стартап, що став провідним сервісом доставки в Латинській Америці, включаючи доставку продуктів харчування.

- b) Cornershop (by Uber) - платформа персональних покупок, придбана Uber, що активно розвивається в Мексиці, Чилі та Перу.

- Близький Схід – emerging markets. ОАЕ та Саудівська Аравія перебувають у стрімкому розвитку:

- Careem (тепер частина Uber) – розширив свої послуги від перевезень до доставки продуктів харчування в країнах Перської затоки.

- Noon – місцева платформа електронної комерції, що активно розвиває сегмент онлайн-продажу продуктів харчування.

Серед ключових регіональних відмінностей можна виділити наступні :

- Швидкість доставки – азіатські ринки лідирують у швидкості доставки (10-30 хвилин), європейські фокусуються на якості (1-2 години), американські на зручності (same-day delivery).

- Технологічні рішення – Азія впроваджує найбільш інноваційні технології (роботизація, AI), Європа фокусується на сталості, США на масштабованості.

- Споживчі переваги – свіжі продукти домінують в Азії, органічні в Європі, зручність та різноманітність в США.

- Бізнес-моделі – азія схиляється до супер-аплікацій, Європа до унікальності, США до інтеграції з існуючими ритейлерами [11].

Ці регіональні особливості створюють унікальні вимоги до систем рекомендацій, які повинні враховувати культурні, технологічні та логістичні специфіки кожного ринку для забезпечення максимальної ефективності та користувацького досвіду.

Незважаючи на динамічне зростання ринку онлайн-торгівлі продуктами харчування, галузь стикається з низкою серйозних викликів, які перешкоджають досягненню повного потенціалу та впливають на прибутковість і сталість бізнесу. Ці проблеми мають як операційний, так і стратегічний характер.

Однією із найкритичніших проблем є досягнення прибутковості в онлайн-торгівлі продуктами харчування. Аналіз показує, що традиційна модель збирання замовлень з полиць магазину більш ніж удвічі збільшує трудові витрати порівняно з звичайними покупцями та руйнує всю прибутковість. Ця проблема особливо гостро постає для компаній, які намагаються конкурувати на ціні, оскільки низькі маржі на продукти харчування не залишають простору для покриття додаткових операційних витрат [12].

Високі витрати на логістику, особливо для доставки холодильних товарів, створюють додатковий тиск на прибутковість. Компанії змушені інвестувати в дорогу інфраструктуру холодного ланцюга поставок, не завжди отримуючи відповідну віддачу від інвестицій у короткостроковій перспективі.

Холодне зберігання та логістика є особливо складними аспектами E-Commerce продуктів харчування. Необхідність підтримання належної температури протягом усього ланцюга поставок – від складу до порогу споживача – вимагає значних інвестицій в спеціалізоване обладнання та інфраструктуру.

Транспортні затримки, нестача робочої сили та зростання вартості сировини ще більше загострюють ці проблеми, в кінцевому рахунку впливаючи на стратегії ценоутворення та прибуткові марж для продуктових

рітейлерів. Особливо це стосується скоропсувних товарів, де будь-яке порушення температурного режиму може призвести до значних втрат.

Пандемія коронавірусу на початку 2020 року виявила багато слабких місць, які може зазнати онлайн-торгівля продуктами в стресових ситуаціях, таких як нездатність веб-сайтів справлятися з піковими навантаженнями. Це підкреслює критичну важливість масштабованої ІТ-інфраструктури, здатної витримувати раптові сплески попиту.

Особливими загрозами є [13]:

- Нестабільність веб-платформ під час високого навантаження.
- Неефективність систем управління запасами в реальному часі.
- Складність інтеграції різних технологічних рішень.
- Потребу в постійному оновленні та модернізації систем.
- Конкурентний тиск та диференціація.

Жорстка конкуренція є одним з основних викликів онлайн-рітейлу на наступний рік. У 2023 році існувало 2,4 мільйона онлайн-магазинів, а до 2024 року ця цифра зростає до 2,55 мільйона - збільшення на 153,000 нових магазинів лише у Сполучених Штатах. Це означає, що щодня відкривається приблизно 419 нових онлайн-магазинів. У такому насиченому середовищі компанії стикаються з наступними проблемами:

- Необхідності виділятися серед конкурентів.
- Підтримання конкурентоспроможних цін при збереженні якості.
- Побудови лояльності клієнтів в умовах легкості переходу до конкурентів.
- Інвестування у маркетинг та брендинг для утримання позицій на ринку.
- Виклики споживчого досвіду.
- Проблема довіри та якості. Споживачі часто не довіряють якості продуктів, особливо свіжих та скоропсувних товарів, при онлайн-покупках. Відсутність можливості фізично оглянути продукт перед покупкою створює

бар'єр для багатьох потенційних клієнтів. Онлайн-рітейлери повинні мати справу з мінливим споживчим попитом, високими показниками повернень та технологіями платежів, що розвиваються. Вони також борються за те, щоб виділитися на переповненому ринку, забезпечуючи при цьому безперебійну логістику та швидку доставку [14].

Технології відіграють вирішальну роль у вирішенні викликів ланцюга поставок продуктів харчування шляхом покращення різних аспектів операцій. Детальна аналітика допоможе брендам визначати точні обсяги виробництва та місця, де інвентар потрібен найбільше. Це додатково усуне непотрібний час транзиту та забезпечить доставку точно в строк для гарантування максимальної свіжості.

Виклики у галузі E-Commerce продуктів харчування є багатоаспектними та взаємопов'язаними. Успішне подолання цих проблем вимагає комплексного підходу, що включає технологічні інновації, оптимізацію операційних процесів, інвестиції в інфраструктуру та розвиток ефективних стратегій залучення клієнтів. Системи рекомендацій у цьому контексті можуть відігравати ключову роль у вирішенні багатьох з цих викликів, особливо в аспектах персоналізації, підвищення конверсії та оптимізації користувацького досвіду.

1.2. Сутність рекомендаційних систем та їх роль в електронній комерції

Рекомендаційні системи стали невід'ємною частиною сучасної електронної комерції, радикально змінюючи спосіб взаємодії споживачів з онлайн-платформами та суттєво впливаючи на бізнес-показники компаній. У контексті онлайн-торгівлі продуктами харчування ці системи набувають особливого значення через специфіку споживчої поведінки та високу частоту повторних покупок [15].

Рекомендаційна система (Recommender System) – це інформаційна система, яка використовує алгоритми машинного навчання та аналіз даних для автоматичного генерування персоналізованих рекомендацій товарів, послуг або контенту для користувачів на основі їхніх переваг, поведінки та характеристик. Рекомендаційні системи допомагають користувачам знаходити відповідні товари серед величезної кількості доступних варіантів, що особливо критично для платформ з широким асортиментом продуктів харчування.

Основна мета рекомендаційних систем полягає у вирішенні проблеми «інформаційного перевантаження», коли користувач стикається з надмірною кількістю варіантів вибору, що ускладнює прийняття рішень про покупку. У контексті продуктів харчування це особливо актуально, оскільки сучасні онлайн-супермаркети можуть пропонувати десятки тисяч товарних позицій.

Основні типи рекомендаційних систем:

- Фільтрація за вмістом (Content-Based Filtering). Цей підхід базується на аналізі характеристик товарів та порівнянні їх з профілем користувача. Системи контентної фільтрації рекомендують товари, схожі на ті, які користувач вже купував або оцінював позитивно. Для продуктів харчування це може включати [16]:

- a) Категоризацію за типом продукту (молочні, м'ясні, овочі).
- b) Поживну цінність (калорійність, вміст білків, жирів, вуглеводів).
- c) Бренд та виробника.
- d) Ціновий сегмент.

- Колаборативна фільтрація (Collaborative Filtering). ґрунтується на принципі, що користувачі з подібними смаками та поведінкою матимуть схожі переваги в майбутньому. Цей метод аналізує поведінку та переваги користувачів для виявлення схожих споживчих патернів. Розрізняють два основні підвиди:

- User-Based Collaborative Filtering – знаходить користувачів з подібними перевагами та рекомендує товари, які сподобались схожим користувачам. Для продуктів харчування це може означати рекомендації на основі схожих дієтичних звичок або кулінарних уподобань.

- Item-Based Collaborative Filtering – аналізує зв'язки між товарами на основі того, як часто їх купують разом різні користувачі. Цей підхід особливо ефективний для продуктів харчування через природні комбінації інгредієнтів та комплементарні товари.

- Гібридні рекомендаційні системи поєднують кілька підходів для подолання обмежень окремих методів. У контексті продуктів харчування найбільш ефективними виявляються комбінації контентної та колаборативної фільтрації для балансу між персоналізацією та відкриттям нових товарів, демографічних даних з поведінковими патернами для врахування регіональних кулінарних традицій, контекстної інформації (час дня, сезон, особливі події) з історією покупок

- Системи на основі знань (Knowledge-Based Systems) використовують експертні знання про предметну область для генерування рекомендацій. Для продуктів харчування це може включати кулінарні знання про поєднання інгредієнтів, дієтологічну інформацію про поживну цінність, сезонність продуктів та регіональні особливості кухні.

- Контекстні рекомендаційні системи враховують ситуаційні фактори при генеруванні рекомендацій. Для продуктів харчування контекст може включати час дня та прийоми їжі, погодні умови, особливі події, географічне розташування користувача, сезонність продуктів [17].

Серед технологій машинного навчання в рекомендаційних системах застосовують класичні алгоритми:

- Matrix Factorization – техніка розкладу матриці «користувач-товар» для виявлення латентних факторів, що впливають на вподобання.

Особливо ефективна для великих наборів даних про покупки продуктів харчування.

- k-Nearest Neighbors (k-NN) – алгоритм пошуку найближчих сусідів для знаходження схожих користувачів або товарів. Простий у реалізації та інтерпретації, що важливо для бізнес-користувачів.

- Clustering Algorithms – методи кластеризації для групування користувачів з подібними перевагами або товарів з подібними характеристиками.

Також виділяють і сучасні підходи на базі глибокого навчання:

- Deep Neural Networks – глибокі нейронні мережі можуть моделювати складні нелінійні залежності у поведінці користувачів, що особливо корисно для розуміння складних харчових переваг.

- Autoencoders – використовуються для зменшення розмірності та виявлення латентних представлень користувачів та товарів.

- Recurrent Neural Networks (RNN/LSTM) – ефективні для моделювання послідовної поведінки користувачів, що важливо для прогнозування повторних покупок продуктів харчування.

- Graph Neural Networks – моделюють складні взаємозв'язки між користувачами, товарами та контекстною інформацією у вигляді графових структур.

Роль рекомендаційних систем у підвищенні продажів і залученні користувачів полягає в наступному:

- Вплив на ключові бізнес-метрики.
- Збільшення конверсії та продажів. Рекомендаційні системи можуть збільшити конверсію до 30% та підвищити середній чек на 10-25%. У контексті продуктів харчування цей вплив особливо помітний через Cross-selling ефект, Up-selling можливості – пропозиції преміум-версій звичних продуктів, імпульсні покупки. Статистика показує, що 35% доходу Amazon

генерується через їхню рекомендаційну систему, що демонструє величезний потенціал цієї технології [18].

- Підвищення утримання клієнтів. Персоналізовані рекомендації створюють більш залучений користувацький досвід, що призводить до збільшення часу, проведеного на платформі, вищої частоти повторних візитів, зниження відсотка відмов (bounce rate), формування лояльності до бренду. Для продуктів харчування це особливо важливо, оскільки споживачі роблять покупки регулярно, і утримання клієнта має довгострокову цінність.

- Оптимізація користувацького досвіду, адже рекомендаційні системи вирішують ключову проблему онлайн-покупок продуктів харчування – складність навігації у великому асортименті. Вони скорочують час пошуку потрібних товарів, допомагають відкривати нові продукти, адаптуються до змін у вподобаннях користувача, враховують дієтичні обмеження та алергії.

- Психологічні аспекти впливу рекомендацій – зменшення когнітивного навантаження. Рекомендаційні системи зменшують когнітивне навантаження на користувачів, спрощуючи процес прийняття рішень. Це особливо важливо для продуктів харчування, де споживачі часто стикаються з «парадоксом вибору» через надмірну кількість опцій.

- Ефект соціального підтвердження – рекомендації типу «інші користувачі також купували» використовують психологічний ефект соціального підтвердження, що підвищує довіру до товару та ймовірність покупки [19].

- Створення відчуття індивідуальності – персоналізовані рекомендації створюють у користувачів відчуття, що платформа «розуміє» їхні потреби, що підвищує лояльність та задоволеність сервісом.

- Бізнес-переваги для операторів платформ.

- Оптимізація управління запасами.

Рекомендаційні системи допомагають прогнозувати попит на товари, оптимізувати розподіл запасів, зменшувати втрати від псування швидкопсувних продуктів, планувати закупівлі та промоакції.

Підвищення операційної ефективності досягається шляхом автоматизації процесу продажів, зменшенням навантаження на службу підтримки клієнтів, оптимізацією маркетингових витрат, покращенням рекламних кампаній. Серед конкурентних переваг можна виділити те, що ефективна рекомендаційна система стає бар'єром для входу конкурентів, джерелом диференціації на ринку, інструментом утримання клієнтів та основою для розвитку додаткових сервісів.

Сучасні платформи демонструють, що рекомендаційні системи стали критично важливим компонентом успішних платформ електронної комерції, особливо у сфері продуктів харчування, де персоналізація може суттєво вплинути на задоволеність клієнтів та бізнес-результати.

1.3. Основні алгоритми сучасних рекомендаційних систем

Колаборативна фільтрація (Collaborative Filtering) базується на аналізі поведінки користувачів і знаходженні схожих уподобань між ними. Основна ідея полягає в тому, що користувачі зі схожими смаками в минулому будуть мати схожі переваги і в майбутньому.

Типи колаборативної фільтрації:

- User-based Collaborative Filtering – знаходить користувачів зі схожими вподобаннями, рекомендує товари, які подобаються схожим користувачам, використовує метрики схожості: косинусна подібність, кореляція Пірсона, евклідова відстань.

- Item-based Collaborative Filtering – аналізує схожість між товарами на основі оцінок користувачів, рекомендує товари, схожі на ті, що

вже подобаються користувачу, більш стабільний, оскільки характеристики товарів змінюються рідше.

До переваг колаборативної фільтрації можна віднести те, що вона:

- Не потребує аналізу змісту товарів.
- Може виявляти неочевидні зв'язки.
- Ефективна для доменів з багатьма користувачами.

Недоліки колаборативної фільтрації:

- Проблема «холодного старту» для нових користувачів/товарів.
- Розрідженість даних.
- Схильність до популярних товарів [20].

Контентна фільтрація (Content-based Filtering) аналізує характеристики товарів і профіль користувача для створення рекомендацій. Система рекомендує товари, схожі на ті, які користувач оцінив позитивно в минулому.

Його принципи роботи полягають у наступному:

1. Профіль користувача:

- Створюється на основі історії взаємодій.
- Містить ваги для різних характеристик товарів.
- Може включати явні (оцінки) та неявні (перегляди, покупки)

дані.

2. Профіль товару:

- Векторне представлення характеристик товару.
- Може включати метадані, текстові описи, зображення.
- Використовує методи NLP для текстового контенту.

3. Методи аналізу контенту:

- TF-IDF для текстових даних.
- Векторизація характеристик.
- Машинне навчання для класифікації.
- Обробка природної мови для текстів.

Серед переваг можна виділити відсутність проблеми холодного старту для товарів, прозорість рекомендацій, персоналізація на основі індивідуальних переваг [21].

До недоліків можна віднести обмежене різноманіття рекомендацій, складність аналізу деяких типів контенту, потреба в якісних метаданих

Гібридні моделі поєднують різні підходи для подолання недоліків окремих методів і підвищення якості рекомендацій. Серед типів гібридних підходів виділяють:

4. Зважену комбінацію, яка має наступні характеристики:
 - Лінійну комбінацію оцінок різних алгоритмів.
 - Ваги, які можуть бути фіксованими або адаптивними.
 - Проста у реалізації, але потребує налаштування вагів.
5. Змішування (Mixed):
 - Презентація рекомендацій з різних алгоритмів одночасно.
 - Користувач бачить результати всіх підходів.
 - Збільшує різноманітність рекомендацій.
6. Перемикання (Switching):
 - Вибір алгоритму залежно від ситуації.
 - Наприклад, контентна фільтрація для нових товарів.
 - Колаборативна фільтрація для активних користувачів.
7. Каскадна модель:
 - Послідовне застосування алгоритмів.
 - Перший алгоритм створює список кандидатів.
 - Другий алгоритм уточнює рейтинг.
8. Мета-рівень (Meta-level):
 - Один алгоритм навчається на виході іншого.
 - Контентна фільтрація може навчити колаборативну.
 - Складніший, але потенційно більш ефективний.

РОЗДІЛ 2 АНАЛІЗ ІСНУЮЧИХ РЕКОМЕНДАЦІЙНИХ СИСТЕМ У СФЕРІ ОНЛАЙН-ПРОДАЖУ ПРОДУКТІВ

2.1. Огляд популярних сервісів (Amazon Fresh, Instacart, Rozetka та інші)

З метою глибшого розуміння сучасної практики впровадження рекомендаційних систем у сфері онлайн-торгівлі продуктами харчування було здійснено аналіз провідних міжнародних та національних платформ, що вже реалізували відповідну функціональність. У даному підрозділі розглядаються чотири типові приклади – Amazon Fresh, Instacart, Rozetka та Glovo – як представники різних підходів до організації персоналізованої взаємодії з користувачем.

Amazon Fresh є дочірнім сервісом корпорації Amazon, який спеціалізується на доставці продуктів харчування та побутових товарів. Він інтегрований з основною платформою Amazon та орієнтований на мешканців мегаполісів США, Європи та Японії. Сервіс пропонує понад пів мільйона товарних позицій, включаючи свіжі овочі, фрукти, м'ясо, заморожені продукти, органічні продукти та товари власних торгових марок. Доставка можлива вже протягом 1–2 годин, що досягається завдяки оптимізованим логістичним маршрутам та високому ступеню автоматизації складів [22].

Instacart – американська платформа, що працює за моделлю агрегатора, об'єднуючи в єдиному інтерфейсі сотні партнерських роздрібних мереж (Walmart, Costco, Kroger тощо). Особливістю сервісу є використання персональних "шоперів", які самостійно комплектують замовлення в офлайн-магазинах та здійснюють доставку. Instacart активно інтегрує сервіси з плануванням меню, дієтичними налаштуваннями користувача та функціоналом рекомендацій на основі кошика покупок [23].

Rozetka є провідною українською мультикатегорійною платформою електронної комерції, що з 2020 року активно розвиває напрямок продажу

продуктів харчування. Каталог включає як імпортовані продукти довготривалого зберігання, так і локальні бренди. Розділ із продуктами харчування функціонує у партнерстві з великими національними мережами (АТБ, METRO, Varus), використовуючи як модель доставки, так і самовивозу з фізичних точок.

Glovo – міжнародна платформа доставки «на вимогу», яка поєднує доставку страв із ресторанів, аптек, супермаркетів та невеликих магазинів. У продуктивній категорії компанія пропонує два варіанти: партнерські магазини (АТБ, Сільпо) та власні “темні” склади Glovo Market. Особливістю сервісу є акцент на швидкості доставки (часто в межах 30 хвилин), що вимагає максимальної ефективності рекомендаційної логіки [24].

Кожен із розглянутих сервісів реалізує унікальний підхід до рекомендацій, обумовлений їхньою бізнес-моделлю, масштабом та доступом до даних. Розглянемо детальніше функціональні аспекти.

Amazon Fresh використовує одну з найпотужніших рекомендаційних інфраструктур, створених у світовій практиці E-Commerce. Її основними компонентами є:

- гібридна модель (поєднання колаборативної та контентної фільтрації);
- постійна адаптація в реальному часі на основі сесійної поведінки;
- голосові рекомендації через Alexa, які дають змогу інтерактивно формувати кошик;
- урахування сезонності, локації, свята, погоди;
- прогнозування повторних замовлень із врахуванням інтервалів між покупками.

Instacart зосереджується на персоналізації у контексті поточної сесії та історії покупок:

- формування так званих Smart Lists на основі регулярних товарів;
- рекомендації з урахуванням дієтичних фільтрів та алергенів;

- пропозиція альтернатив у разі відсутності товару;
- інтеграція з рецептами, що автоматично додають усі необхідні інгредієнти до кошика;
- адаптація під конкретну торгову мережу користувача.

Rozetka реалізує базову систему рекомендацій, спрямовану переважно на підвищення середнього чека [25]:

- популярні товари на основі історії переглядів та замовлень;
- перехресні рекомендації ("з цим товаром також купують");
- промо-блоки з сезонними або акційними позиціями;
- персоналізація з використанням куки-файлів та геоданих.

Glovo фокусується на швидких та релевантних рекомендаціях у рамках «моментального споживання»:

- контекстно-залежні підказки (час доби, день тижня);
- рекомендації «на основі попередніх замовлень»;
- пропозиції готових рішень (наприклад, «вечеря на двох», «перекус»);
- використання AI для предиктивного аналізу поведінки на короткій дистанції.

На основі проведеного аналізу можна зробити висновок, що Amazon Fresh та Instacart мають найбільш розвинені та гнучкі системи рекомендацій, що забезпечують глибоку персоналізацію та адаптивність. Rozetka дотримується більш традиційного підходу, компенсуючи технологічну складність розгалуженою категоризацією та UX. Glovo успішно імплементує рекомендації в умовах високої швидкості доставки, орієнтуючись на короткотривалі споживчі патерни [26].

Врахування функціонального досвіду цих платформ дозволяє чітко визначити вимоги до проектування рекомендаційної системи для українського продуктового ринку, адаптованої до реалій швидкопсувнього асортименту, локальної логістики та змінних уподобань користувачів.

2.2 Аналіз сильних і слабких сторін реалізованих рішень

Ефективність рекомендаційної системи не обмежується лише точністю алгоритмів. Вона формується в комплексі таких факторів, як якість даних, архітектура взаємодії, масштабованість, адаптивність до поведінки користувача та інтегрованість у бізнес-процеси. В даному підрозділі систематизовано сильні та слабкі сторони проаналізованих платформ, а також розглянуто вплив кожного аспекту на загальний користувацький досвід.

Такі сервіси як Amazon Fresh та Instacart демонструють високий рівень UX-зрілості, оскільки їхні інтерфейси інтуїтивні, логіка взаємодії оптимізована під повторювані дії користувачів, а рекомендації органічно інтегровані в сесію покупки. Amazon адаптує порядок і тип рекомендацій відповідно до етапу формування кошика, а Instacart динамічно оновлює пропозиції, орієнтуючись на контекст сесії. Рекомендації подаються ненав'язливо, часто у формі підказок, пов'язаних із діями користувача.

Алгоритмічна складність і глибина Amazon Fresh використовує багаторівневу гібридну архітектуру: поєднання колаборативної фільтрації (CF), контентного аналізу (CBF), контекстно-залежної логіки (CARS) та глибокого навчання. Інфраструктура побудована на основі масштабованих нейромережових моделей (наприклад, BERT4Rec, DIN). Instacart використовує сегментовану фільтрацію з урахуванням геолокації, звичок, переваг і реального асортименту магазинів-партнерів. Обидва сервіси застосовують предиктивні моделі для прогнозування регулярних замовлень [27].

Платформи демонструють високий ступінь інтегрованості. Зокрема, Amazon Fresh має глибоку взаємодію з Alexa (голосові запити), Smart Home-пристроями та мобільними застосунками. Instacart інтегрує рецепти, списки покупок, дієтичні фільтри та замовлення за історією. Це формує безшовний користувацький досвід і підвищує ймовірність повторного використання.

Масштабованість та адаптація до локальних ринків полягає у тому, що глобальні сервіси демонструють чудові результати в мегаполісах, їх важко адаптувати до місцевих ринків з обмеженою інфраструктурою або нестабільним асортиментом. Рекомендації можуть стати нерелевантними в разі частих змін на складах партнерських магазинів (типово для Instacart). Також платформи часто не враховують регіональні дієтичні традиції та локальні бренди [28].

У багатьох системах виникає проблема початкової нерелевантності, коли платформа не має достатньої кількості даних для персоналізації. Це особливо критично для Rozetka та Glovo, де рекомендації часто ґрунтуються на загальній популярності, а не на індивідуальних інтересах. Без впровадження додаткових механізмів (опитувальники, інтерактивна сегментація) Cold Start-сценарії призводять до низької конверсії нових користувачів [29].

У деяких випадках рекомендації подаються занадто настирливо або дублюють вже обрані товари, що викликає фрустрацію у користувачів. Також частою є ситуація, коли рекомендовані товари або не входять до наявного асортименту, або не відповідають поточному контексту (наприклад, рекомендація морозива взимку). Glovo, орієнтуючись на швидкість, іноді нехтує якістю персоналізації, обмежуючись загальними шаблонами.

Сильні та слабкі сторони рекомендаційних систем мають прямий вплив на ключові показники ефективності платформи, зокрема:

- Конверсію – релевантні, ненав'язливі та динамічно оновлювані рекомендації підвищують ймовірність покупки, особливо у випадку повторних замовлень (Amazon, Instacart).

- Частоту повернення користувача – системи, які «вчаться» з часом і демонструють зростання релевантності, формують довіру і сприяють лояльності.

– Середній чек – крос-продажі, засновані на історичних або контекстуальних шаблонах, дозволяють ефективно збільшувати загальний обсяг покупки (особливо через “рецептурні набори” Instacart).

– Фрустрацію та відтік – нерелевантні, повторювані чи помилкові рекомендації знижують задоволення від користування сервісом, що особливо критично для нових клієнтів з обмеженим терпінням до недосконалих алгоритмів [30].

Таким чином, проектування та реалізація рекомендаційної системи повинні враховувати не лише технічну реалізованість моделей, але й контекст, у якому рекомендації подаються. UX, семантична відповідність та інфраструктурна масштабованість мають спільний вплив на загальний рівень довіри користувача до платформи.

2.3 Порівняння підходів до рекомендацій

Різні онлайн-платформи застосовують різні підходи до побудови рекомендаційних систем залежно від доступних даних, масштабу бізнесу, цільової аудиторії та технічної інфраструктури. У цьому підрозділі здійснюється системне порівняння застосованих моделей та принципів їх реалізації на прикладі чотирьох ключових гравців – Amazon Fresh, Instacart, Rozetka та Glovo [31].

Порівняння було проведено на основі наступних критеріїв:

- тип застосованих алгоритмів та їхню складність;
- підходи до обробки даних і персоналізації;
- інтеграцію з інтерфейсом користувача;
- ефективність рекомендацій у контексті UX і комерційної конверсії;
- гнучкість до масштабування та адаптації [32].

Порівняльну характеристику сервісів, згідно наведених критеріїв представлено в табл. А.1 додатка А. Як видно з даної таблиці, найвищу інженерну складність і гнучкість демонструють гібридні системи, такі як у Amazon. Натомість більш локальні або агреговані сервіси (Glovo, Rozetka) використовують спрощені логіки, орієнтовані на мінімізацію обчислювальних ресурсів і швидкість взаємодії [33].

Усі сервіси вирішують одну й ту ж задачу – персоналізацію вибору в умовах великого асортименту, але шляхом різного балансу між точністю, швидкістю, масштабованістю та глибиною моделі. Ці спостереження ляжуть в основу формування вимог до власної рекомендаційної системи в наступному розділі.

Аналіз ефективності алгоритмів рекомендаційної логіки базується на оцінці як кількісних показників (точність, повнота, NDCG, MAP), так і якісних аспектів – впливу на поведінку користувача, зростання конверсій, підвищення задоволеності тощо. У численних наукових публікаціях зазначається, що гібридні моделі демонструють стабільно високі результати в e-commerce сценаріях, де дані користувача є достатньо повними [34].

Згідно з дослідженням компанії Amazon (Zhou et al., 2019), впровадження алгоритму Deep Neural Collaborative Filtering (DNMF) призвело до зростання коефіцієнта кліків (CTR) на 25%, а середній чек збільшився на 12%. Аналогічно, застосування Context-Aware Recommendation Systems (Adomavicius & Tuzhilin, 2015) дозволяє підвищити релевантність у сценаріях із обмеженим часом взаємодії, що характерно для Glovo.

У 2021 році Instacart опублікувала результати використання Item2Vec-інспірованої моделі, яка забезпечила до 8% приросту точності рекомендацій у повторних покупках. Особливо ефективною вона виявилася при формуванні автоматизованих списків покупок (smart carts) [35].

Загалом, порівняння традиційних підходів (наприклад, popularity-based) з нейромережевими (deep learning) демонструє перевагу останніх у складних

середовищах, де доступні великі обсяги даних та потрібно адаптуватися до поведінкових змін користувачів у режимі реального часу [36].

Узагальнення проведеного аналізу дозволяє виділити низку ключових факторів, які мають бути враховані при проектуванні ефективної рекомендаційної системи для онлайн-продажу продуктів харчування:

1. Модульність та гібридність, оскільки архітектура майбутньої системи повинна підтримувати інтеграцію кількох моделей – від класичної колаборативної фільтрації до адаптивного контентного аналізу, а в перспективі – глибинного навчання [37].

2. Контекстна адаптивність, адже необхідно враховувати часові параметри, геолокацію, сезонність та інші фактори, що впливають на купівельну поведінку. Особливо це важливо у food-сфері, де патерни можуть суттєво змінюватись залежно від дня тижня чи погоди.

3. М'який старт для нових користувачів. Cold-start проблема повинна вирішуватись через гібридні методи сегментації, попередні опитувальники або стартові сценарії (наприклад, «рекомендовані набори») [38].

4. Інтеграція з бізнес-логікою, оскільки система рекомендацій повинна враховувати залишки на складах, поточні акції, промо-набори та логістичні обмеження – це підвищить релевантність і зменшить фрустрацію користувача [39].

5. UX-інтегрованість, адже рекомендації мають подаватися ненав'язливо, адаптивно до фази сесії, з можливістю швидкої реакції користувача (наприклад, додавання до кошика одним кліком, фільтри алергенів, дієт, типів кухні тощо) [40].

Таким чином, створення власної системи повинно поєднувати гнучкість, масштабованість та високий рівень персоналізації – із чітким фокусом на користувацький досвід, простоту взаємодії та актуальність рекомендацій у режимі реального часу.

РОЗДІЛ 3 ФУНКЦІОНАЛЬНІ ТА НЕФУНКЦІОНАЛЬНІ ВИМОГИ ДО МАЙБУТНЬОЇ СИСТЕМИ

3.1. Визначення основних функцій системи

Формулювання функціональних вимог до системи рекомендацій базується на врахуванні специфіки предметної області, аналізі сучасних реалізацій та очікуваної поведінки кінцевого користувача. Усі функції, необхідні для ефективного функціонування системи, можна умовно поділити на дві групи: обов'язкові та додаткові. Також доцільним є опис типових сценаріїв використання, які дозволяють уявити роботу системи в динаміці.

Обов'язкові функції формують основу логіки системи та є критичними для реалізації персоналізованих рекомендацій [41]. До них належать компоненти, без яких система не зможе виконувати своє призначення. Обов'язкові функції системи рекомендацій показано в табл. А.2 додатка А.

Ці функції реалізують базову персоналізацію та є невід'ємною частиною будь-якої сучасної рекомендаційної системи.

Також можна виділити і додаткові функції, такі як персоналізація, інтеграція з обліковими записами [42]. Вони покращують гнучкість системи, розширюють користувацький досвід та дозволяють адаптувати рекомендації до більш індивідуальних запитів. Додаткові функції системи рекомендацій представлені в табл. А.3 додатка А.

Реалізація цих функцій дозволяє системі не лише адаптуватися до різних типів користувачів, а й активно впливати на підвищення повторних продажів, середнього чека та загальної лояльності [43]. Користувацькі сценарії описують типові випадки взаємодії користувача із системою рекомендацій на онлайн-платформі продажу продуктів харчування. Вони допомагають визначити, як саме користувачі можуть отримувати персоналізовані рекомендації, і які функції система має підтримувати для забезпечення

зручності та ефективності. Основні користувацькі сценарії, які передбачено реалізувати, наведені в таблиці А.4 додатка А.

Реалізація даних сценаріїв дозволяє значно покращити користувацький досвід, оскільки забезпечує релевантність інформації та зручність взаємодії з платформою [44]. При цьому важливо забезпечити, щоб система реагувала на зміни в поведінці користувача в режимі реального часу або з мінімальною затримкою.

Крім того, кожен із сценаріїв вимагає певного набору функціональних компонентів системи, таких як збір та обробка даних, модулі рекомендацій, інтерфейс користувача, а також компоненти безпеки для захисту персональних даних [45].

Для забезпечення стабільної роботи та комфортного використання майбутньої системи рекомендацій у сфері онлайн-продажу продуктів харчування необхідно врахувати важливі нефункціональні вимоги. Вони охоплюють три основні аспекти: продуктивність системи, її здатність масштабуватися при зростанні обсягів даних і користувачів, а також якість користувацького досвіду (UX).

Продуктивність системи визначає швидкість реакції на запити користувачів і генерації персоналізованих рекомендацій. Висока продуктивність критично важлива для збереження інтересу клієнтів та забезпечення безперебійної роботи платформи.

3.2. Вимоги до продуктивності, масштабованості та UX

Масштабованість дозволяє системі ефективно обробляти зростаючі обсяги інформації, враховуючи збільшення кількості товарів, транзакцій та користувачів. Це вимагає продуманої архітектури з підтримкою горизонтального та вертикального масштабування.

UX-дизайн відіграє ключову роль у залученні та утриманні користувачів, оскільки зручний і адаптивний інтерфейс сприяє позитивному враженню від використання платформи та підвищує лояльність клієнтів.

Продуктивність рекомендаційної системи визначається швидкістю обробки запитів та генерації рекомендацій, що безпосередньо впливає на загальний досвід користувача [46]. В онлайн-торгівлі продуктами харчування критично важливо забезпечити мінімальні затримки, особливо під час пікових навантажень, коли кількість одночасних користувачів значно зростає. Для оптимізації продуктивності застосовуються кілька основних підходів [47].

По-перше, використання ефективних алгоритмів рекомендацій дозволяє швидко обробляти великі обсяги даних, мінімізуючи час відповіді системи [48]. По-друге, кешування результатів популярних запитів допомагає зменшити навантаження на обчислювальні модулі, забезпечуючи швидке повернення часто запитуваної інформації [49].

Також важливим є розподіл обчислень між серверними компонентами, що дає змогу паралельно обробляти множину запитів. Нарешті, застосування сучасних технологій баз даних, оптимізованих для швидкого пошуку та вибірки інформації, значно підвищує загальну продуктивність системи [50]. У таблиці А.5 додатка А наведено основні методи оптимізації продуктивності. Вони є ключовим фактором забезпечення високої якості роботи системи рекомендацій, що позитивно впливає на користувацький досвід і рівень задоволеності клієнтів.

Масштабованість системи рекомендацій є важливим нефункціональним показником, що визначає здатність платформи ефективно обробляти зростаючі обсяги даних та збільшення кількості користувачів без втрати продуктивності.

У контексті онлайн-торгівлі продуктами харчування масштабованість має особливе значення, оскільки обсяг інформації про товари, транзакції та поведінку користувачів зростає експоненційно. Система повинна

підтримувати обробку великих обсягів даних про продукти, їх характеристики, наявність і ціни, аналіз історії покупок та поведінкових патернів мільйонів користувачів, а також генерацію рекомендацій у реальному часі навіть при значних навантаженнях.

Для забезпечення масштабованості застосовують різні підходи. Горизонтальне масштабування передбачає додавання нових серверів для розподілу навантаження між кількома вузлами. Вертикальне масштабування – це збільшення потужності існуючого обладнання, зокрема CPU та RAM. Використання розподілених баз даних дозволяє розподілити дані між кількома серверами для підвищення швидкості доступу та обробки інформації. Крім того, мікросервісна архітектура розділяє систему на незалежні компоненти, що дає змогу гнучко масштабувати окремі модулі.

У таблиці А.6 додатка А наведено основні підходи до забезпечення масштабованості. Застосування цих методів дає змогу гарантувати безперебійну роботу системи навіть у періоди пікових навантажень та підтримувати високу якість рекомендацій для широкої аудиторії користувачів

Якісний UX-дизайн є одним із ключових аспектів успішної рекомендаційної системи, оскільки саме через інтерфейс користувач взаємодіє з платформою і приймає рішення про покупку. Адаптивність інтерфейсу забезпечує коректне відображення та функціональність на різних пристроях, таких як комп'ютери, смартфони та планшети, що дозволяє охопити ширшу аудиторію. Важливим аспектом є також доступність, що передбачає підтримку користувачів із різними фізичними обмеженнями, зокрема через використання екранних читалок, висококонтрастних кольорових схем та інших інструментів для полегшення взаємодії.

Зручність навігації реалізується за допомогою інтуїтивно зрозумілих меню і пошуку, які дозволяють швидко знаходити потрібні товари і рекомендації. Візуальна привабливість інтерфейсу досягається за рахунок збалансованого поєднання кольорів, шрифтів та графічних елементів, що

підвищує залученість користувачів та робить перебування на сайті комфортнішим. Ефективні UX-рішень у рекомендаційних системах наведені в табл. А.7 додатка А. Добре продуманий UX-дизайн сприяє підвищенню задоволеності користувачів, збільшує час перебування на платформі та позитивно впливає на конверсію.

3.3. Проблеми безпеки та приватності користувацьких даних

У сучасних рекомендаційних системах, особливо в галузі онлайн-торгівлі продуктами харчування, питання безпеки та приватності користувацьких даних є надзвичайно важливими. Збір, обробка та зберігання персональної інформації створюють потенційні ризики витоку даних, несанкціонованого доступу та зловживань. Втрата довіри з боку користувачів через порушення безпеки може призвести до суттєвих репутаційних і фінансових втрат.

Забезпечення безпеки вимагає комплексного підходу, що охоплює як технічні заходи захисту, так і дотримання законодавчих норм щодо приватності. Крім того, важливо забезпечити прозорість у процесах збору та обробки даних, інформуючи користувачів про те, як саме їх інформація використовується, та надаючи можливість контролювати власні дані. Далі розглянемо основні загрози безпеці, з якими можуть стикатися рекомендаційні системи.

Рекомендаційні системи піддаються різноманітним загрозам, які можуть суттєво вплинути на їхню працездатність і безпеку користувачів. Серед основних загроз варто виділити наступні:

- втручання у дані – зловмисники можуть змінювати або видаляти дані про користувачів чи товари, що призводить до некоректної роботи алгоритмів рекомендацій і втрати довіри користувачів;

- зломи акаунтів – неавторизований доступ до облікових записів користувачів дозволяє отримувати персональні дані, історію покупок та інші чутливі відомості;

- атаки на систему – DDoS-атаки можуть призвести до недоступності сервісу, що негативно впливає на користувацький досвід.

- шахрайство – використання фальшивих облікових записів і ботів для маніпулювання рейтингами товарів і рекомендаціями. Класифікацію основних загроз наведено у табл. А.8 додатка А. Врахування цих загроз є необхідною умовою для розробки надійної та безпечної системи рекомендацій.

Захист персональних даних у рекомендаційних системах регулюється низкою міжнародних та національних законів і стандартів, серед яких найвідомішим є Загальний регламент захисту даних (GDPR), прийнятий Європейським Союзом. Цей регламент встановлює жорсткі вимоги до збору, зберігання, обробки та передачі персональних даних користувачів з метою забезпечення їхньої приватності та контролю над інформацією.

GDPR передбачає право користувачів знати, які саме дані про них збираються і як вони використовуються, можливість вимагати виправлення або видалення своїх персональних даних, а також необхідність отримання чіткої згоди користувача на обробку персональної інформації. Впроваджуються заходи для запобігання несанкціонованому доступу та втраті даних.

Крім GDPR, у різних країнах діють свої закони щодо захисту персональних даних, такі як Закон України «Про захист персональних даних» або Закон США «California Consumer Privacy Act», які також потрібно враховувати при розробці рекомендаційної системи для забезпечення юридичної відповідності.

Для відповідності цим нормам рекомендується реалізовувати механізми прозорості політики конфіденційності, забезпечувати можливість управління

користувачем своїми даними та використовувати стандарти безпечної обробки інформації.

Таким чином, дотримання законодавчих вимог є не лише правовою необхідністю, а й складовою частиною формування довіри користувачів до платформи, що безпосередньо впливає на її успіх і репутацію.

Для забезпечення безпеки персональних даних у рекомендаційних системах застосовуються різноманітні технічні методи захисту, які дозволяють мінімізувати ризики несанкціонованого доступу та витоку інформації.

Шифрування є одним із основних засобів захисту даних. Воно забезпечує конфіденційність інформації під час її передачі і зберігання, перетворюючи дані у формат, недоступний для розуміння без відповідного ключа розшифрування. Використання сучасних алгоритмів шифрування, таких як AES або RSA, дозволяє значно знизити ймовірність компрометації даних.

Анонімізація даних полягає у видаленні або заміні ідентифікуючих ознак користувача, що робить неможливим встановлення особистості з наявної інформації. Цей метод особливо корисний для аналізу великих обсягів даних без порушення приватності.

Крім того, застосовуються механізми контролю доступу, багатофакторна автентифікація, моніторинг подій безпеки та регулярне оновлення програмного забезпечення для усунення вразливостей.

Впровадження комплексного підходу до захисту даних, що поєднує шифрування, анонімізацію та інші заходи, забезпечує високий рівень безпеки і сприяє довірі користувачів до рекомендаційної системи. Основні методи захисту персональних даних у рекомендаційних системах показано в табл. А.9 додатка А.

Для забезпечення безпеки персональних даних у рекомендаційних системах застосовуються різноманітні технічні методи захисту, які

дозволяють мінімізувати ризики несанкціонованого доступу та витоку інформації.

Шифрування є одним із основних засобів захисту даних. Воно забезпечує конфіденційність інформації під час її передачі і зберігання, перетворюючи дані у формат, недоступний для розуміння без відповідного ключа розшифрування. Використання сучасних алгоритмів шифрування, таких як AES або RSA, дозволяє значно знизити ймовірність компрометації даних.

Анонімізація даних полягає у видаленні або заміні ідентифікуючих ознак користувача, що робить неможливим встановлення особистості з наявної інформації. Цей метод особливо корисний для аналізу великих обсягів даних без порушення приватності.

Крім того, застосовуються механізми контролю доступу, багатофакторна автентифікація, моніторинг подій безпеки та регулярне оновлення програмного забезпечення для усунення вразливостей.

Впровадження комплексного підходу до захисту даних, що поєднує шифрування, анонімізацію та інші заходи, забезпечує високий рівень безпеки і сприяє довірі користувачів до рекомендаційної системи.

РОЗДІЛ 4 ПРОЄКТУВАННЯ СТРУКТУРИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

4.1. Сценарії використання (Use Case діаграми)

Сценарії використання є фундаментальним інструментом для визначення функціональних вимог до рекомендаційної системи та моделювання взаємодії між різними типами користувачів і системою. У контексті онлайн платформи з продажу продуктів харчування, правильно спроектовані сценарії використання забезпечують чітке розуміння бізнес-потреб та технічних вимог до системи.

Рекомендаційна система для платформи електронної комерції продуктів харчування має унікальні особливості, пов'язані зі специфікою товарів: терміни придатності, сезонність, харчові вподобання та дієтичні обмеження користувачів. Це вимагає особливого підходу до проектування функціональності та взаємодії з користувачами.

Для ефективного функціонування рекомендаційної системи визначено три основні типи акторів, кожен з яких має специфічні потреби та права доступу:

1. Незареєстрований користувач (Guest User)^

- Представляє потенційних покупців, які вперше відвідують платформу.
- Має обмежений доступ до функціональності.
- Основна мета – ознайомлення з асортиментом та прийняття рішення про реєстрацію.
- Потребує базових рекомендацій для формування первинного інтересу.

2. Зареєстрований користувач (Registered User):

- активні користувачі платформи з власним профілем.
- Мають доступ до персоналізованих функцій.

- Генерують дані про взаємодію з системою.
- Основна цільова аудиторія для рекомендаційних алгоритмів.

3. Адміністратор системи (System Administrator)

- Відповідає за налаштування та моніторинг системи
- Має доступ до аналітичних даних та інструментів управління
- Забезпечує оптимізацію роботи рекомендаційних алгоритмів

Загальна діаграма сценаріїв використання показана в додатку Б.1

Неzareєстровані користувачі складають значну частину трафіку електронних платформ і потребують особливої уваги для конверсії в активних покупців. Система повинна надавати їм достатньо інформації та рекомендацій для прийняття рішення про покупку або реєстрацію. Діаграма сценарію незареєстрованих користувачів відображена на рис. 4.1.

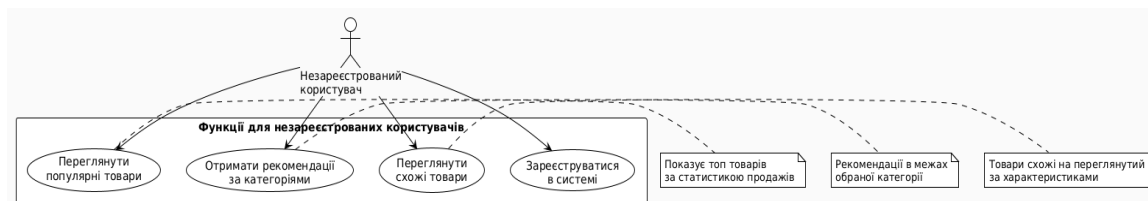


Рисунок 4.1 – Діаграма сценарію незареєстрованих користувачів

Система електронної комерції передбачає кілька ключових сценаріїв взаємодії користувача, зокрема перегляд популярних товарів, отримання персоналізованих рекомендацій у межах категорій, а також ознайомлення зі схожими позиціями на основі вже переглянутого товару.

Перший сценарій стосується перегляду популярних товарів. Незареєстрований користувач, перебуваючи на головній сторінці платформи, автоматично отримує доступ до рейтингу найбільш популярних товарів. Система самостійно формує цей список, аналізуючи статистику продажів, кількість переглядів, середню оцінку товарів за останні 30 днів. Після обробки даних система відображає топ-20 товарів у зручному форматі, що дає можливість користувачеві ознайомитися з деталями кожної позиції. У разі

тимчасової недоступності даних система відображає попередньо збережений перелік.

Другий сценарій – це отримання рекомендацій за категоріями. Після того, як користувач обирає певну товарну категорію, система формує список відповідних товарів, фільтруючи їх за рейтингом і наявністю. Всі пропозиції сортуються за комбінованим показником популярності та оцінок користувачів. Таким чином, відображається добірка актуальних та релевантних товарів, яку можна додатково фільтрувати. Якщо обрана категорія не містить товарів, система пропонує альтернативні варіанти.

Третій сценарій полягає у відображенні схожих товарів. Коли користувач переглядає сторінку певного товару, система аналізує його характеристики – бренд, категорію, ціновий сегмент – і знаходить подібні товари. Результати сортуються за ступенем схожості, після чого користувачеві пропонується добірка з 5–10 альтернатив, що відповідають його інтересам. Такий підхід дозволяє покращити користувацький досвід, підвищити ймовірність покупки та забезпечити гнучку систему навігації в межах онлайн-платформи. Діаграма послідовності для незареєстрованих користувачів зображена в додатку Б.2

Система електронної комерції також активно працює з зареєстрованими користувачами, які становлять ключову аудиторію для впровадження персоналізованих рекомендацій. Завдяки збору історії покупок, аналізу вподобань та моделюванню поведінкових патернів, платформа може формувати точні та релевантні пропозиції, які відповідають індивідуальним інтересам користувача.

У випадку отримання персональних рекомендацій система активується, коли користувач авторизується та переходить до особистого кабінету або персональної сторінки. Після завантаження профілю та історії його взаємодій із платформою, система обирає оптимальну модель для генерації рекомендацій. Якщо наявна повна історія покупок, застосовується алгоритм

коллаборативної фільтрації, який враховує подібність до інших користувачів. Якщо дані обмежені, система використовує контентно-орієнтований підхід у поєднанні з найпопулярнішими товарами. Рекомендації ранжуються за релевантністю й новизною та виводяться разом з коротким поясненням, чому саме ці товари запропоновано. Після взаємодії з рекомендаціями модель користувача оновлюється, що підвищує точність майбутніх прогнозів. Блок-схема персональних рекомендацій для зареєстрованих користувачів показана в додатку Б.3

Ще одним важливим сценарієм є рекомендації на основі вмісту кошика. Коли користувач додає товар у кошик, система аналізує цей вміст і застосовує алгоритм «Market Basket Analysis» для пошуку супутніх товарів, які часто купуються разом. На основі асоціативних правил система виявляє товари з високим рівнем співіснування, перевіряє їхню доступність та відповідність інтересам користувача і пропонує їх у вигляді блоку «Часто купують разом». Ці товари можна додати до кошика швидко та зручно одним кліком, що сприяє збільшенню середнього чека та покращенню користувацького досвіду. Діаграма сценарію зареєстрованих користувачів зображена на рис. 4.2.

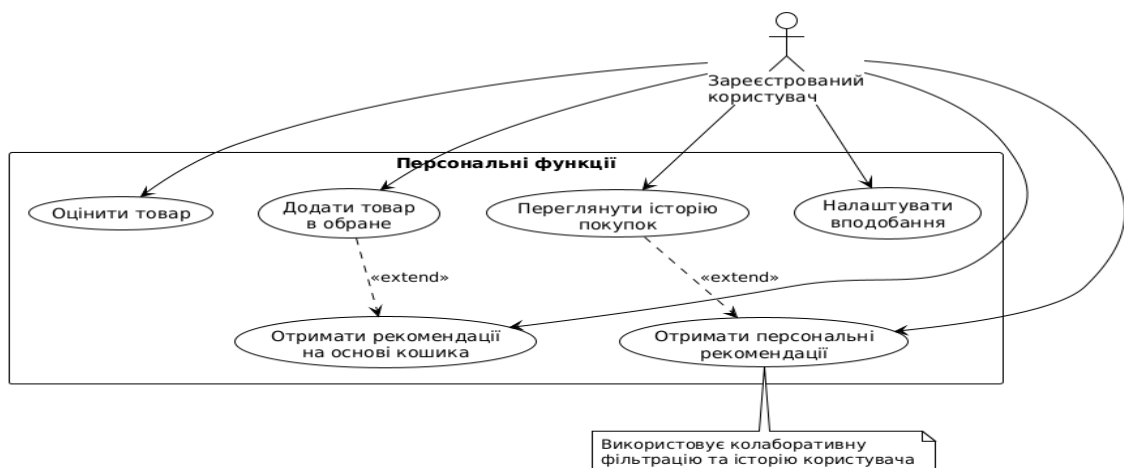


Рисунок 4.2 – Діаграма сценарію зареєстрованих користувачів

Система рекомендацій у платформі електронної комерції не лише орієнтована на користувача, а й включає стратегічний рівень управління, що забезпечується через адміністративний інтерфейс. Діаграму адміністративних сценаріїв зображено на рис.4.3. Адміністративні функції відіграють важливу роль у забезпеченні якості, ефективності та розвитку рекомендаційних алгоритмів. Однією з ключових задач адміністратора є аналіз ефективності системи рекомендацій.

Після входу до аналітичної панелі адміністратор отримує доступ до статистики за обраний період. Система автоматично збирає дані про взаємодію користувачів із рекомендаціями: кількість переглядів, кліків, доданих у кошик товарів, завершених покупок тощо. На основі цих даних розраховуються основні метрики: CTR (Click-Through Rate), що показує, скільки користувачів переходять за рекомендаціями; Conversion Rate – частка покупок серед тих, хто переглянув рекомендовані товари; а також точність (Precision@K) і повнота (Recall@K) рекомендацій у топ-K результатах. Додатково аналізується охоплення (Coverage), різноманітність (Diversity) та новизна (Novelty) рекомендацій, що дозволяє оцінити не лише технічну якість алгоритму, а й його здатність пропонувати різнопланові й актуальні товари.

Система формує порівняльні звіти з попередніми періодами, генерує дашборди та візуалізації, які допомагають швидко виявити тренди, зміни в поведінці користувачів або потенційні проблеми. Після аналізу адміністратор отримує автоматизовані рекомендації для оптимізації моделей: наприклад, зміну вагових коефіцієнтів у гібридній моделі, посилення новизни у видачі або розширення охоплення категорій. Діаграма активності для аналізу ефективності показана в додатку Б.4. Таким чином, платформа забезпечує замкнений цикл самонавчання, де результативність рекомендацій постійно поліпшується завдяки зворотному зв'язку та динамічному управлінню. Матриця трасування функціональних вимог показана в додатку А.11.

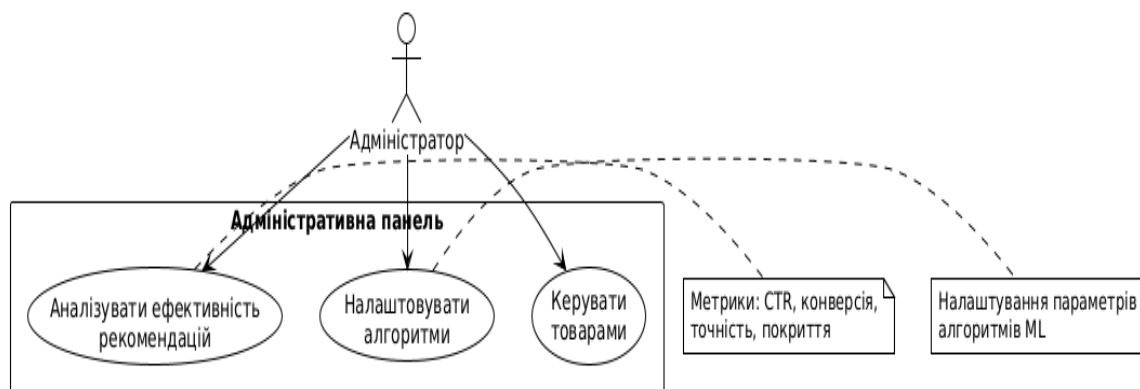


Рисунок 4.3 – Діаграма адміністративних сценаріїв

Аналіз нефункціональних вимог до рекомендаційної системи дозволяє оцінити її здатність функціонувати ефективно, безпечно та стабільно в умовах реального навантаження. Основні аспекти охоплюють продуктивність, надійність, масштабованість та безпеку.

У контексті продуктивності система має забезпечити швидкий час відгуку – не більше 2 секунд для 95% запитів. Вона повинна підтримувати обробку до 10 тисяч одночасних користувачів без зниження якості сервісу. Окрему увагу приділено динамічному оновленню рекомендацій – дані повинні актуалізуватися в режимі реального часу, що забезпечує релевантність видачі.

Надійність системи передбачає високий рівень доступності – не менше 99,9% на рік, що означає менш ніж 9 годин простою. У разі збоїв відновлення сервісу не повинно перевищувати 15 хвилин. Регулярне резервне копіювання кожні 6 годин дозволяє запобігти втраті важливих даних.

Масштабованість системи забезпечується завдяки горизонтальному масштабуванню, що дозволяє додавати обчислювальні ресурси відповідно до зростання навантаження. Підтримка мікросервісної та розподіленої архітектури створює гнучке середовище для розвитку. Особливо важливою є можливість впровадження нових типів рекомендацій без необхідності зміни основної архітектури, що робить систему легко розширюваною.

Безпека охоплює шифрування персональних даних користувачів, контроль доступу на основі ролей (RBAC) та логування всіх адміністративних

дій. Це дозволяє гарантувати конфіденційність, цілісність даних і прозорість управлінських процесів.

Підсумовуючи, розроблені сценарії використання охоплюють ключові аспекти функціонування системи для різних груп користувачів. Незареєстровані користувачі отримують базові рекомендації з метою залучення до активнішої взаємодії з платформою. Зареєстровані користувачі користуються перевагами персоналізації, що підвищує лояльність і задоволеність. Адміністратори мають інструменти для стратегічного аналізу ефективності, що сприяє постійному вдосконаленню системи.

Використання PlantUML для моделювання дозволяє підтримувати діаграми у зрозумілому вигляді з використанням стандартної UML нотації. Це забезпечує легкість інтеграції в системи контролю версій та сприяє прозорості комунікації між членами команди розробки.

У контексті архітектурного проектування система представлена у вигляді контекстної діаграми, де вона взаємодіє із зовнішніми сутностями: користувачами, адміністраторами, зовнішніми API, платіжними системами та веб-додатком електронної комерції. Вся інформація централізовано обробляється й передається у відповідні компоненти: систему аналітики, модулі рекомендацій, бази даних. Такий підхід створює гнучку, надійну й масштабовану архітектуру, що готова до подальшого розвитку та інтеграції нових можливостей.

4.2. Схема взаємодії компонентів (Data Flow Diagram)

Схема взаємодії компонентів визначає архітектуру рекомендаційної системи та потоки даних між її основними модулями. Система побудована за принципом мікросервісної архітектури для забезпечення масштабованості та надійності.

Контекстна діаграма показує рекомендаційну систему як єдине ціле у взаємодії з зовнішніми сутностями. Вона демонструє основні потоки даних між системою та її оточенням без деталізації внутрішньої структури. Дана діаграма зображена в додатку Б.5.

У контекстній діаграмі рекомендаційна система зображена як центральний елемент, що взаємодіє з кількома ключовими зовнішніми сутностями та системами. Взаємодія відбувається через обмін даними, запитами та відповідями, що забезпечує повноцінне функціонування всієї екосистеми платформи.

Користувач – кінцевий споживач системи, який взаємодіє з інтерфейсом платформи. Він генерує поведінкові дані через перегляди товарів, кліки по рекомендаціях, додавання до кошика та покупки. На основі цієї активності система формує персоналізовані рекомендації, які повертаються користувачу в адаптованому форматі.

Адміністратор – відповідає за налаштування алгоритмів, моніторинг ефективності системи та прийняття рішень щодо її подальшої оптимізації. Отримує доступ до аналітичних звітів, дашбордів і метрик, що відображають результативність рекомендаційних моделей.

Зовнішні API – джерела інформації про товари, ціни, наявність на складах та інші параметри, що впливають на релевантність рекомендацій. Дані з API постійно оновлюються, дозволяючи системі враховувати актуальний контекст.

До зовнішніх систем, що забезпечують повноцінну взаємодію, належать:

- Веб-додаток електронної комерції – основне середовище, у якому відбувається користувацька взаємодія з платформою. Саме тут інтегровано рекомендаційні блоки, персональні добірки та супутні товари.

- Система аналітики – інструмент збору та аналізу даних про ефективність рекомендацій. Вона обчислює ключові метрики, такі як CTR,

Conversion Rate, Precision@K тощо, і формує аналітичну основу для ухвалення рішень.

– Платіжна система – надає інформацію про фіналізацію транзакцій, що дозволяє системі оцінювати результативність рекомендацій не лише за переглядами, а й за фактом покупки, формуючи повний цикл взаємодії "перегляд → рекомендація → покупка".

– Такий підхід до побудови архітектури створює гнучке, інтегроване середовище для забезпечення якісного та персоналізованого користувацького досвіду.

Діаграма нульового рівня (Level 0 DFD) показана в додатку Б.6, деталізує внутрішню логіку функціонування рекомендаційної системи, розкриваючи п'ять ключових процесів, які взаємодіють між собою та з відповідними сховищами даних. Вона дозволяє побачити, як система трансформує вхідні дані у цінний результат – персоналізовані рекомендації.

Першим етапом є процес збору та обробки користувацьких даних (1.0). Цей модуль відповідає за отримання інформації з різних джерел: дії користувачів на платформі, дані з зовнішніх API, налаштування профілю, історія покупок. Отримані дані проходять валідацію, очищення та нормалізацію перед тим, як потрапити до відповідних сховищ: база даних користувачів (D1), база взаємодій (D3) та аналітичне сховище (D5).

Другим є процес генерації рекомендацій (2.0) – аналітичне ядро системи, де застосовуються алгоритми колаборативної та контентної фільтрації, гібридні підходи, а також асоціативний аналіз. Цей процес тісно взаємодіє з базою товарів (D2), базою взаємодій (D3) та кешем рекомендацій (D4), де зберігаються попередньо згенеровані результати для швидкої відповіді на запити.

Далі система переходить до персоналізації та фільтрації (3.0), де з урахуванням профілю користувача, його поточної активності та уподобань адаптується кінцева вибірка рекомендацій. Тут відбувається фільтрація за

наявністю товарів, новизною, рейтингом або іншими параметрами, а результати оптимізуються перед відправкою у фронтенд.

Аналітика та моніторинг (4.0) – це процес, який забезпечує зворотній зв'язок. Він відповідає за збір метрик ефективності: CTR, конверсії, точність і повноту рекомендацій, охоплення та різноманітність. Дані зберігаються в аналітичному сховищі (D5) і використовуються як для звітності, так і для подальшої оптимізації системи.

Завершує цю структуру процес управління системою (5.0). Це адміністративний модуль, через який адміністратор може змінювати налаштування алгоритмів, порогові значення метрик, обирати типи рекомендацій або запускати тестування нових моделей.

Загальна структура системи забезпечує логічну, масштабовану та надійну взаємодію між модулями. Завдяки розподілу на окремі процеси й сховища, кожен компонент може бути вдосконалений окремо, не порушуючи цілісності системи. Такий підхід особливо ефективний для підтримки великих обсягів даних і реалізації персоналізованого досвіду в умовах високого навантаження.

Діаграма першого рівня (Level 1 DFD) показана в додатку Б.7 та деталізує критичний для системи процес генерації рекомендацій, який є основою персоналізованого користувацького досвіду. Цей процес поділяється на п'ять взаємопов'язаних підпроцесів, що реалізують різні підходи до формування рекомендацій, забезпечуючи їхню точність, релевантність і адаптивність.

Перший підпроцес – колаборативна фільтрація (2.1) – базується на пошуку подібності між користувачами або товарами. Алгоритми типу User-based CF, Item-based CF та методи факторизації матриць використовуються для обчислення схожості, після чого формується матриця подібності, яка лягає в основу рекомендацій. Цей підхід ефективний за умови наявності достатньої кількості даних про взаємодії.

Другий підпроцес – контентна фільтрація (2.2) – працює з властивостями товарів. За допомогою TF-IDF аналізуються текстові описи, а cosine similarity використовується для порівняння векторних уявлень. У результаті формуються профілі товарів, які можна зіставляти з уподобаннями користувача, навіть якщо історія його покупок є обмеженою.

Третій підпроцес – гібридний алгоритм (2.3) – інтегрує результати двох попередніх підходів. Він зважує внесок кожного методу, компенсує їхні слабкі сторони й підвищує точність рекомендацій завдяки поєднанню знань про поведінку користувача та контент товарів.

Четвертий підпроцес – генерація популярних рекомендацій (2.4) – активується в умовах відсутності історичних даних (ефект «холодного старту»). Він ґрунтується на загальній популярності товарів, враховуючи тренди, сезонність і загальну активність на платформі, дозволяючи залучити нових користувачів через релевантні загальні пропозиції.

Завершальним є підпроцес агрегації результатів (2.5), який об'єднує всі типи рекомендацій. На цьому етапі застосовуються стратегії змішування, ранжування та фільтрації, що дозволяє сформувати фінальний список рекомендацій. Цей список передається далі на етап персоналізації, де він адаптується під конкретного користувача в реальному часі.

Такий рівень деталізації дає змогу глибше зрозуміти архітектуру обробки даних та алгоритмічну основу системи, забезпечуючи як точність рекомендацій, так і їхню гнучкість для різних сценаріїв використання.

Архітектура рекомендаційної системи побудована на принципах мікросервісної моделі, що забезпечує гнучкість, масштабованість та стійкість до збоїв. Діаграма архітектури рекомендаційної системи показана в додатку А.8. Такий підхід дозволяє розділити функціональність системи на окремі, логічно завершені сервіси, кожен із яких виконує чітко визначену бізнес-функцію та може розгортатися незалежно від інших.

Основу архітектури складають чотири ключові принципи:

По-перше, слабка зв'язаність означає, що сервіси обмінюються інформацією через стандартизовані API, не втручаючись у внутрішню логіку один одного. Це знижує ризики при оновленнях і дозволяє легше масштабувати окремі компоненти.

По-друге, висока згуртованість передбачає, що кожен сервіс відповідає за окрему бізнес-ділянку – наприклад, керування користувачами, каталог товарів, аналітика або рекомендації.

По-третє, незалежне розгортання надає можливість оновлювати сервіси поетапно без зупинки всієї системи, що критично для безперервної роботи.

І, нарешті, відмовостійкість гарантує, що збій одного компонента не порушить роботу інших, а система загалом продовжить функціонувати, ізолюючи проблеми.

На чолі всієї взаємодії стоїть API Gateway – єдина точка входу для всіх клієнтських запитів. Вона відповідає за маршрутизацію до потрібних сервісів, контроль швидкості запитів (rate limiting), аутентифікацію, логування та моніторинг.

Серед мікросервісів виділяються:

- User Service – обробка профілів користувачів, реєстрація, аутентифікація.
- Product Service – зберігання та оновлення інформації про товари.
- Interaction Service – фіксація поведінкових дій користувачів (кліки, покупки, оцінки).
- Recommendation Service – побудова рекомендацій на основі моделей.
- Analytics Service – обробка метрик, формування звітів.
- Configuration Service – централізоване управління параметрами системи.

Кожен сервіс використовує оптимізоване сховище даних. Наприклад, PostgreSQL підходить для обліку користувачів, MongoDB – для динамічного каталогу товарів, ClickHouse – для аналітики великих обсягів подій, Redis – як швидкий кеш для збереження готових рекомендацій, а InfluxDB – для часових метрик.

Для обробки даних у реальному часі використовується Kafka – платформа потокової обробки, що дозволяє обробляти події по мірі їх надходження. У свою чергу, Apache Spark забезпечує пакетну обробку великих історичних даних, а ML Pipeline реалізує повний цикл навчання, тестування та оновлення моделей машинного навчання, що використовуються для генерації рекомендацій.

Завдяки такій архітектурі система залишається гнучкою, адаптивною до змін, легко масштабованою та готовою до розширення без необхідності глибоких змін у кодовій базі. Вона ідеально відповідає вимогам сучасних систем електронної комерції з високим навантаженням і вимогами до персоналізації.

Система рекомендацій побудована з урахуванням підтримки обробки подій у реальному часі, показана в додатку Б.9, що є критично важливим для актуальності персональних пропозицій та швидкої адаптації до змін у поведінці користувачів. Такий підхід дозволяє системі реагувати на взаємодію миттєво, формуючи рекомендації, які відповідають поточним інтересам кожного користувача.

Процес обробки в реальному часі розпочинається з збору подій: кожен клік, перегляд товару або покупка фіксується на клієнтському рівні. Отримані дані проходять валідацію та нормалізацію, що дозволяє відсіяти помилки, привести події до уніфікованого формату та підготувати до подальшої обробки.

Наступний крок – публікація події в Kafka. Це дозволяє асинхронно передавати дані між мікросервісами, не навантажуючи основний потік

користувацької взаємодії. Recommendation Service отримує ці події через механізм споживання Kafka-топіків.

Після надходження нової події система перевіряє кеш у Redis: чи існують уже готові рекомендації для відповідного користувача або подібної ситуації. Якщо кеш застарів або відсутній, ініціюється процес генерації нових рекомендацій, який використовує дані події в режимі реального часу.

Після побудови персонального списку рекомендацій результати зберігаються в кеші, що забезпечує їхню швидку доступність при повторному запиті. Паралельно запускається асинхронне навчання – модель машинного навчання оновлюється у фоновому режимі з урахуванням нових даних, що гарантує постійне вдосконалення рекомендаційної логіки.

Основними перевагами такої обробки є: по-перше, миттєва реакція на будь-яку активність користувача дозволяє оперативно пропонувати актуальні товари. По-друге, система адаптується до змін трендів і попиту в режимі реального часу, що критично важливо в умовах динамічного ринку. По-третє, це значно покращує користувацький досвід, адже рекомендації виглядають доречними й своєчасними. І нарешті, така оперативність сприяє збільшенню конверсій, оскільки користувач отримує релевантні пропозиції в момент максимальної зацікавленості.

Таким чином, обробка подій у реальному часі забезпечує не лише технологічну гнучкість системи, а й безпосередньо впливає на її бізнес-ефективність.

Матриця відповідальності компонентів (RACI) визначає ролі та відповідальність кожного мікросервісу для основних функцій системи. Це забезпечує чітке розуміння того, хто відповідає за виконання завдань, хто підзвітний за кінцевий результат, хто консультується під час виконання, а хто лише інформується про результати.

Визначення ролей:

- Responsible – відповідальний за виконання завдання

- Accountable – підзвітний за кінцевий результат
- Consulted – консультиується під час виконання
- Informed – інформується про результати

Таблиця 4.1 – Матриця відповідальності компонентів

Функція	User Service	Product Service	Recommendation Service	Analytics Service
Управління профілями	R,A	I	C	I
Каталог товарів	I	R,A	C	I
Генерація рекомендацій	C	C	R,A	I
Збір метрик	I	I	C	R,A
Персоналізація	C	I	R,A	C
A/B тестування	I	I	C	R,A

У системі:

- User Service виконує ключову роль у керуванні користувачами, несе відповідальність за обробку профілів та надає консультаційну підтримку іншим сервісам у питаннях, пов’язаних із користувачами.
- Product Service відповідає за ведення каталогу товарів, контролює весь життєвий цикл товарної інформації та забезпечує доступ до неї для інших компонентів системи.
- Recommendation Service виступає центральним елементом системи персоналізації, відповідає за генерацію рекомендацій і несе основну відповідальність за їхню релевантність і точність.
- Analytics Service зосереджується на зборі, обробці та аналізі даних, надає аналітичні звіти, інсайти та результати експериментів, що використовуються для покращення інших сервісів.

Система використовує три основні підходи до обробки даних, кожен з яких оптимізований для окремих сценаріїв використання.

Пакетна обробка застосовується для роботи з великими обсягами історичних даних та періодичного оновлення моделей машинного навчання.

Вона виконується за розкладом, наприклад щоденно або щотижня, охоплює всю доступну історичну інформацію, оптимізована для максимальної пропускної здатності та забезпечує високу точність моделей. Потік обробки виглядає так: історичні дані передаються до Apache Spark, далі здійснюється обробка та навчання моделі, результати оновлюють рекомендації, які кешуються для подальшого використання.

Потокова обробка відповідає за реєстрацію та обробку подій користувачів у режимі реального часу. Вона забезпечує мінімальну затримку, адаптується до поточної поведінки користувача та дозволяє інкрементально оновлювати рекомендації. Потік подій проходить через Kafka, надходить у компонент потокової обробки, де формується актуальна рекомендація, що подається через API.

У випадку холодного старту застосовуються спеціальні стратегії, спрямовані на нових користувачів, які ще не мають історії взаємодій. Серед таких стратегій: формування рекомендацій на основі демографічних даних (вік, стать, місцезнаходження), популярності товарів у категоріях, які потенційно цікавлять користувача, врахування явно вказаних інтересів, а також поступове навчання на основі перших взаємодій. Потік включає обробку демографічних даних, формування базових рекомендацій і поступову персоналізацію.

Для оптимізації продуктивності система використовує кілька технік. Кешування дозволяє зберігати попередньо розраховані рекомендації у Redis, причому кеш оновлюється асинхронно при зміні даних. Тривалість життя кешу налаштовується відповідно до типу рекомендацій.

Індексація передбачає створення індексів для пришвидшеного пошуку схожих товарів та користувачів. Застосовуються як класичні, так і векторні індекси для семантичного пошуку, що регулярно оновлюються для збереження актуальності даних.

Горизонтальне масштабування забезпечується через розподіл навантаження між інстансами сервісів, використання шардінгу баз даних і впровадження механізмів автоматичного масштабування відповідно до поточного навантаження.

Завдяки такій багаторівневій архітектурі обробки даних система досягає високої продуктивності, гнучкості, масштабованості та стійкості до збоїв, а також легко адаптується до нових алгоритмів і типів рекомендацій відповідно до потреб бізнесу.

4.3 Попередня модель зберігання та обробки даних

Модель зберігання та обробки даних, показаної в додатку Б.10, визначає загальну архітектуру зберігання, структуру баз даних, схеми організації інформації та підходи до її обробки в межах рекомендаційної системи. У системі використовується полігібридний підхід, який поєднує реляційні, документоорієнтовані та векторні СУБД. Це дозволяє гнучко працювати з різними типами даних – транзакційними, напівструктурованими, аналітичними та семантичними – і забезпечувати як високу продуктивність, так і масштабованість відповідно до потреб системи.

Концептуальна модель даних сформована відповідно до принципів domain-driven design (DDD), що дає змогу відобразити бізнес-логіку системи в чітко визначених доменних сутностях. Основними доменами є користувачі, товари, події взаємодії, сесії, рекомендації, інтереси тощо. Для кожної сутності визначено атрибути, зв'язки та контексти використання. Логічна модель даних показана в додатку А.11.

Оперативні дані організовано у нормалізованій формі, що забезпечує цілісність, уніфікованість і зручність оновлення. У той же час для аналітичних запитів і генерації рекомендацій використовуються денормалізовані представлення, які агрегують інформацію в зручному для швидкого доступу

вигляді. Це дозволяє ефективно виконувати обчислення, фільтрацію та персоналізацію в реальному часі або у пакетному режимі.

Таким чином, концептуальна модель поєднує стабільну основу для зберігання критичних даних із гнучкою інфраструктурою для машинного навчання та аналітики.

SQL схема для PostgreSQL відображена в додатку В.1.

MongoDB використовується у системі для зберігання слабоструктурованих, динамічних або часто змінюваних даних, де реляційна модель була б занадто жорсткою або неефективною. Завдяки схемофлексивній природі, MongoDB ідеально підходить для зберігання інформації, структура якої може відрізнятись від документа до документа, а також для швидкого доступу до вкладених об'єктів.

Основні типи даних, що зберігаються в MongoDB:

- Каталог товарів (product catalog): кожен документ описує товар і містить такі поля, як назва, опис, характеристики, атрибути, ціна, доступність, варіації, медіаконтент, теги, відгуки. Структура документа дозволяє вкладати масиви, об'єкти або навіть локалізовані версії описів.

- Контент (content blocks): інформаційні сторінки, банери, промоакції та рекомендаційні блоки, які змінюються в реальному часі залежно від сегментації користувача або бізнес-правил. Гнучкість JSON-структури дозволяє легко оновлювати ці дані без впливу на інші частини системи.

- Профілі товарів і поведінкові атрибути: додаткові властивості товарів, зібрані в процесі взаємодії з користувачами – наприклад, показники переглядів, додавання до кошика, емоційні теги, асоціації з іншими товарами.

MongoDB забезпечує:

- Високу швидкість читання і запису для складених документів
- Можливість швидко оновлювати окремі елементи без рекомпозиції всієї структури

- Потужні механізми пошуку та фільтрації (включаючи повнотекстовий пошук, агрегації)
- Вбудовану підтримку масштабування за допомогою шардингу.

Таким чином, MongoDB виконує роль гнучкого сховища для бізнес-даних, які змінюються динамічно, потребують глибокої вкладеності або персоналізованого підходу до зберігання. JavaScript код для створення індексів MongoDB відображений в додатку В.2.

ВИСНОВКИ

У межах цієї роботи було проведено всебічне дослідження з розробки системи рекомендацій для онлайн-платформи з продажу продуктів харчування. Основною метою стало створення архітектурної концепції, яка враховує специфіку ринку, особливості користувацької поведінки, а також технічні вимоги до масштабованої, персоналізованої та ефективної рекомендаційної системи.

На першому етапі проаналізовано стан онлайн-торгівлі продуктами харчування в Україні та світі. Виявлено, що цей ринок демонструє стабільне зростання, зумовлене зміною споживчих звичок, розвитком електронної комерції та зростанням попиту на персоналізовані цифрові сервіси. Зокрема, на фоні глобальних змін у логістиці, зростає роль автоматизованих рішень, які покращують користувацький досвід і підвищують лояльність покупців. Ці фактори зумовили доцільність впровадження адаптивної системи рекомендацій саме в цій предметній області.

Далі проведено дослідження теоретичних основ побудови рекомендаційних систем. Розглянуто ключові підходи – колаборативну фільтрацію, контент-орієнтовані моделі, а також гібридні стратегії. Встановлено, що для платформи з продуктами харчування найбільш ефективними є гібридні системи, які здатні поєднувати переваги різних методів та долати обмеження кожного окремо взятого підходу (наприклад, проблему «холодного старту» чи надто узагальнених рекомендацій).

Окрему увагу приділено аналізу алгоритмів, які використовуються в сучасних рекомендаційних системах. Визначено, що методи на основі латентного факторного моделювання (наприклад, ALS), кластеризації, дерев рішень, а також нейромережеві підходи можуть бути ефективно застосовані для обробки поведінкових даних користувачів у цій галузі. Розглянуто їхню придатність саме до продуктового ритейлу, де важливу роль відіграють

фактори сезонності, термінів придатності, змін у асортименті та повторюваність покупок.

З метою оцінки стану впровадження рекомендацій у промислових умовах проведено порівняльний аналіз функціональності рекомендаційних систем у провідних онлайн-платформах (Amazon Fresh, Instacart, Rozetka та ін.). Це дозволило виокремити ключові патерни та підходи до персоналізації, а також визначити ті функції, які потребують подальшого вдосконалення – зокрема, адаптивність до змін поведінки користувача в реальному часі та прозорість логіки рекомендацій.

На основі зібраного матеріалу сформульовано чіткі функціональні та нефункціональні вимоги до майбутньої системи. Серед функціональних – підтримка персоналізації, швидкий підбір товарів, обробка великих обсягів подій, адаптація до нового користувача. Серед нефункціональних – масштабованість, продуктивність, інтеграційна сумісність, відмовостійкість і безпека.

Фінальним результатом роботи стало проектування логічної та концептуальної структури системи рекомендацій. Розроблено модель зберігання та обробки даних із полігібридним підходом до вибору СУБД: PostgreSQL для транзакційних даних, MongoDB для слабоструктурованого контенту, а також Redis, Kafka й інші компоненти для обробки потоків і кешування. Враховано сценарії пакетної, потокової та гібридної обробки даних, що забезпечує гнучкість та ефективність при роботі з різними типами навантажень.

Таким чином, поставлені дослідницькі та проєктні завдання виконано повною мірою. Результати роботи можуть бути використані як основа для реалізації практичної системи рекомендацій на онлайн-платформі з продажу продуктів харчування. Запропонована архітектура є масштабованою, адаптивною та орієнтованою на сучасні технологічні стандарти, що відкриває перспективи для її подальшого впровадження та розвитку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Aggarwal C. C. Recommender Systems: The Textbook. Cham Springer, 2017. 497 p. – URL: <https://link.springer.com/book/10.1007/978-3-319-29659-3> (дата звернення: 12.01.2025).
2. Zhang S., Yao L., Sun A., Tay Y. Deep Learning based Recommender System: A Survey and New Perspectives // ACM Computing Surveys. 2019. – Vol. 52(1). P. 1–38. URL: <https://dl.acm.org/doi/10.1145/3285029> (дата звернення: 20.01.2025).
3. Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for Recommender Systems // Computer. 2009. Vol. 42(8). P. 30–37. URL: <https://ieeexplore.ieee.org/document/5197422> (дата звернення: 02.02.2025).
4. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. 2nd ed. Cham: Springer, 2015. 1003 p. URL: <https://link.springer.com/book/10.1007/978-1-4899-7637-6> (дата звернення: 06.02.2025).
5. Zhang Y., Chen X. Explainable Recommendation: A Survey and New Perspectives // Foundations and Trends in Information Retrieval. 2020. Vol. 14(1). P. 1–101. URL: <https://www.nowpublishers.com/article/Details/INR-093> (дата звернення: 20.02.2025)..
6. Grus J. Data Science from Scratch. – 2nd ed. – Sebastopol : O’Reilly Media, 2019. – 406 p. – URL: <https://www.oreilly.com/library/view/data-science-from/9781492041122/> (дата звернення: 20.02.2025).
7. Nielsen N. Practical Recommender Systems. Shelter Island : Manning, 2017. 265 p. URL: <https://www.manning.com/books/practical-recommender-systems> (дата звернення: 27.02.2025).
8. Beel J., Gipp B., Langer S., Breitinger C. Research-paper recommender systems: a literature survey // International Journal on Digital Libraries. 2017.

- Vol. 17. P. 305–338. URL: <https://link.springer.com/article/10.1007/s00799-015-0156-0> (дата звернення: 20.03.2025).
9. Chen L., Pu P. User-centric Evaluation of Recommender Systems: Survey of the State of the Art // User Modeling and User-Adapted Interaction. 2017. Vol. 27(3). P. 167–200. URL: <https://link.springer.com/article/10.1007/s11257-017-9195-2> (дата звернення: 20.03.2025).
10. Covington P., Adams J., Sargin E. Deep Neural Networks for YouTube Recommendations // Proceedings of the 10th ACM Conference on Recommender Systems. 2016. P. 191–198. URL: <https://dl.acm.org/doi/10.1145/2959100.2959190> (дата звернення: 20.03.2025).
11. Microsoft. Recommenders: Examples and Best Practices. 2024. URL: <https://github.com/microsoft/recommenders> (дата звернення: 20.03.2025).
12. Google Cloud. Building a Recommendation System with Vertex AI. 2023. URL: <https://cloud.google.com/vertex-ai/docs/tutorials/recommendation-system> (дата звернення: 27.04.2025).
13. Amazon Personalize Documentation. 2024. URL: <https://docs.aws.amazon.com/personalize/latest/dg/what-is-personalize.html>
14. Instacart. The Instacart Platform and Algorithms. 2023. URL: <https://tech.instacart.com> (дата звернення: 27.04.2025).
15. Rozetka. Інтернет-магазин продуктів харчування. 2024. URL: <https://rozetka.com.ua/ua/groceries/> (дата звернення: 27.04.2025).
16. Statista. Online Food Delivery Worldwide. 2024. URL: <https://www.statista.com/outlook/dmo/online-food-delivery/worldwide> (дата звернення: 27.04.2025).
17. Deloitte. Online Grocery Report 2023. URL: <https://www2.deloitte.com/global/en/pages/consumer-industrial-products/articles/online-grocery.html> (дата звернення: 27.04.2025).

18. Gartner. How to Build an Effective Recommendation Engine. 2023. URL: <https://www.gartner.com/en/documents/4001356> (дата звернення: 29.04.2025).
19. Towards Data Science. A Guide to Building Recommendation Systems. 2022. URL: <https://towardsdatascience.com/a-guide-to-building-recommendation-systems-bd8c8102323> (дата звернення: 29.04.2025).
20. MongoDB. Building a Product Catalog with MongoDB. 2024. URL: <https://www.mongodb.com/blog/post/building-product-catalogs-with-mongodb> (дата звернення: 03.04.2025).
21. Sun F., Liu J., Wu J. et al. Bert4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer // Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM). 2019. P. 1441–1450. URL: <https://dl.acm.org/doi/10.1145/3357384.3357895> (дата звернення: 10.04.2025).
22. Rendle S., Freudenthaler C., Gantner Z., Schmidt-Thieme L. BPR: Bayesian Personalized Ranking from Implicit Feedback // Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI). 2009. P. 452–461. URL: <https://arxiv.org/abs/1205.2618> (дата звернення: 13.04.2025).
23. He X., Liao L., Zhang H. et al. Neural Collaborative Filtering // Proceedings of the 26th International Conference on World Wide Web (WWW). 2017. – P. 173–182. – URL: <https://dl.acm.org/doi/10.1145/3038912.3052569> (дата звернення: 13.04.2025).
24. Zhang M., Yao D., Li S., Hu Y. Product Recommendation with Temporal Dynamics // Information Sciences. 2022. Vol. 597. P. 69–89. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0020025522004796> (дата звернення: 13.04.2025).
25. Lu J., Wu D., Mao M. et al. Recommender System Application Developments: A Survey // Decision Support Systems. 2018. Vol. 110. P. 28–

49. URL:
<https://www.sciencedirect.com/science/article/pii/S0167923618300575>
(дата звернення: 20.04.2025).
26. Netflix Tech Blog. Recommending for 200 Million Members. 2021. URL:
<https://netflixtechblog.com/recommending-for-200-million-members-ceda19a8c7f> (дата звернення: 20.04.2025).
27. Shopify. Personalization for Online Grocery Stores. 2023. URL:
<https://www.shopify.com/blog/grocery-personalization> (дата звернення: 20.04.2025).
28. Elasticsearch. Product Recommendations with Elasticsearch and Machine Learning. 2022. URL:
<https://www.elastic.co/blog/product-recommendations-elasticsearch-ml> (дата звернення: 20.04.2025).
29. AWS. Best Practices for Building a Recommendation Engine Using Amazon Personalize. 2023. URL:
<https://aws.amazon.com/blogs/machine-learning/best-practices-for-building-a-recommendation-engine-using-amazon-personalize> (дата звернення: 03.05.2025).
30. Facebook Engineering. Understanding Instagram's Explore Recommendations. 2022. URL:
<https://engineering.fb.com/2022/06/15/ml-applications/instagram-recommendations/> (дата звернення: 03.05.2025).
31. Alibaba Cloud. Personalized Product Recommendations Using Machine Learning. – 2021. – URL:
https://www.alibabacloud.com/blog/personalized-product-recommendations-using-machine-learning_598355 (дата звернення: 10.05.2025).
32. Medium. Hybrid Recommendation System with Collaborative and Content Filtering. 2020. URL:
<https://medium.com/swlh/hybrid-recommender-systems-collaborative-content-filtering-bdaa802b61b0> (дата звернення: 10.05.2025).

33. Medium. How Instacart Uses AI to Recommend What You Eat. 2021. URL: <https://medium.com/instacart-data-science/how-instacart-uses-ai-to-recommend-what-you-eat-37dc8994e9d5> (дата звернення: 14.05.2025).
34. Coursera. Recommender Systems Specialization (University of Minnesota). 2024. URL: <https://www.coursera.org/specializations/recommender-systems> (дата звернення: 14.05.2025).
35. IBM Cloud. Deploying Scalable Recommender Systems on the Cloud. 2022. URL: <https://www.ibm.com/cloud/blog/scalable-recommender-systems> (дата звернення: 14.05.2025).
36. Kaggle. Instacart Market Basket Analysis Dataset. 2023. URL: <https://www.kaggle.com/competitions/instacart-market-basket-analysis> (дата звернення: 18.05.2025).
37. FastAPI. Building a Recommendation API with FastAPI and Python. 2023. – URL: <https://fastapi.tiangolo.com/tutorial/recommendation-api/> (дата звернення: 18.05.2025).
38. Redis. Real-Time Personalization with Redis and Recommender Systems. – 2022. URL: <https://redis.com/blog/real-time-personalization-recommender/> (дата звернення: 18.05.2025).
39. MLflow. Managing Machine Learning Models for Recommendation Systems. 2023. URL: <https://mlflow.org/docs/latest/tutorials-and-examples.html#recommendation> (дата звернення: 20.05.2025).
40. Hugging Face. Transformers for Sequential Recommendation. 2024. URL: <https://huggingface.co/blog/sequential-recommendation-transformers> (дата звернення: 20.05.2025).
41. Polato I., de Souza A., Rego L. et al. A Survey on Recommender Systems for Online Food Delivery Services // Journal of Intelligent & Fuzzy Systems. 2022. Vol. 43(5). P. 5645–5656. URL: <https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs222117> (дата звернення: 20.05.2025).

42. Ferreira A., Gomes H., Lima F. et al. Food Recommender Systems: A Systematic Review // ACM Computing Surveys. 2022. Vol. 54(10s). P. 1–36. URL: <https://dl.acm.org/doi/10.1145/3477452> (дата звернення: 20.05.2025).
43. Ghazarian S., Nikolaev A., Ghosh R. Preference-based Food Recommender System for People with Diabetes // Procedia Computer Science. 2020. Vol. 170. P. 388–395. URL: <https://www.sciencedirect.com/science/article/pii/S1877050920302968> (дата звернення: 25.05.2025).
44. He Z., Yin H., Chen Y. et al. Online Food Recommendation: Datasets, Methods and Challenges // arXiv preprint. 2020. arXiv:2007.10560. URL: <https://arxiv.org/abs/2007.10560> (дата звернення: 25.05.2025).
45. Uber Eats Engineering. Improving Food Discovery with Recommendations. 2021. URL: <https://www.uber.com/blog/improving-food-discovery/> (дата звернення: 25.05.2025).
46. Databricks. Building Real-Time Recommender Systems with Apache Spark. 2023. URL: <https://www.databricks.com/blog/2023/04/real-time-recommendations-apache-spark.html> (дата звернення: 25.05.2025).
47. Snowflake. Modern Data Architecture for Food Retailers. 2023. URL: <https://www.snowflake.com/blog/data-architecture-grocery-retail/> (дата звернення: 25.05.2025).
48. DVC. Versioning Machine Learning Pipelines for Food Recommendations. 2022. URL: <https://dvc.org/blog/food-recommendation-pipelines> (дата звернення: 30.05.2025).
49. Papers with Code. State-of-the-Art Models for Food Recommendation. 2024. URL: <https://paperswithcode.com/task/food-recommendation>
50. MLOps Community. Best Practices for Deploying Recommendation Models in Production. 2023. URL: <https://mlops.community/deploying-recommender-systems/> (дата звернення: 05.06.2025).

ДОДАТОК А – ТАБЛИЦІ

ДОДАТОК А.1 – Порівняльна характеристика сервісів

№	Платформа	Алгоритмічна модель	Обробка даних	UX інтеграція	Сильні сторони	Обмеження
1	Amazon Fresh	Гібридна модель (CF + CBF + DNN)	Big Data, реальний час, поведінкова аналітика	Глибока персоналізація, Alexa-інтеграція	Висока точність, адаптивність, голосові рекомендації	Висока ресурсомісткість, cold-start для нових користувачів
2	Instacart	Колаборативна + Rule-based + Recipes	Сесійні дані, локальні товари, історія покупок	Рецепт-орієнтовані рекомендації	Контекстна релевантність, інтеграція з реальними магазинами	Обмеженість асортименту, залежність від партнерських даних
3	Rozetka	Popularity-based + базовий CF	Категорії, перегляди, історія замовлень	Стандартні блоки рекомендацій	Простота реалізації, широка база користувачів	Обмежена персоналізація, слабка адаптація до food-сегменту
4	Glovo	Heuristics + Context-aware	Мінімальні профілі, швидкі запити	Миттєві поради, часові шаблони	Швидкість, простота, геолокація	Низька точність, відсутність глибокої персоналізації

Джерело: створено автором на основі даних [21]

ДОДАТОК А.2 – Обов'язкові функції системи рекомендацій

№	Функція	Призначення
1	Генерація персоналізованих рекомендацій	Формування релевантного списку товарів на основі історії покупок, вподобань, поведінки інших користувачів
2	Облік історії покупок і дій	Збір і збереження даних про замовлення, перегляди, додавання до кошика
3	Динамічне оновлення рекомендацій	Адаптація списку рекомендацій при зміні кошика, категорії або поведінки в сесії
4	Фільтрація недоступних товарів	Автоматичне виключення з рекомендацій товарів, що недоступні в регіоні чи на складі

ДОДАТОК А.3 – Додаткові функції системи рекомендацій

№	Функція	Призначення
1	Профілювання користувача	Можливість вручну або автоматично вказати харчові вподобання, обмеження, алергії
2	Автоматичне формування списків	Створення «розумного кошика» на основі регулярних покупок
3	Інтеграція з обліковими записами	Авторизація через Google, Facebook, Apple ID із синхронізацією даних
4	Реакція на зворотний зв'язок	Механізм лайків, дизлайків або приховування товарів, що дозволяє уточнювати модель
5	Повідомлення	Надсилання push- або email-сповіщень про новинки, знижки, регулярні товари

ДОДАТОК А.4 – Основні користувацькі сценарії взаємодії із системою рекомендацій

Сценарій	Опис	Ціль користувача
1. Отримання персоналізованих рекомендацій на головній сторінці	Користувач заходить на платформу і одразу бачить підбірку продуктів, рекомендованих на основі його попередніх покупок та поведінки	Полегшення пошуку цікавих продуктів та підвищення лояльності
2. Рекомендації при пошуку товару	Під час введення запиту система пропонує не лише релевантні товари, але й супутні продукти, які можуть зацікавити користувача	Розширення асортименту та збільшення середнього чеку
3. Персоналізація в кошику	На основі поточного вмісту кошика система пропонує додаткові товари, які доповнюють покупки користувача	Підвищення кількості продажів через крос-продажі
4. Оцінка та відгуки	Користувач може оцінити рекомендовані товари і залишити відгук, що впливатиме на подальші рекомендації	Покращення точності рекомендацій через зворотний зв'язок
5. Управління вподобаннями	Користувач має можливість вказати свої вподобання та заборонені категорії товарів для більш точної персоналізації	Забезпечення індивідуального підходу та зручності

ДОДАТОК А.5 – Основні методи оптимізації продуктивності.

№	Метод оптимізації	Опис
1	Використання ефективних алгоритмів	Алгоритми, що швидко обробляють великі масиви даних
2	Кешування результатів	Збереження популярних відповідей для швидкого повторного використання
3	Розподіл обчислень	Паралельна обробка запитів на різних серверних компонентах
4	Сучасні технології баз даних	Оптимізовані бази даних для швидкого пошуку та вибірки

ДОДАТОК А.6 – Основні підходи до забезпечення масштабованості системи рекомендацій.

№	Підхід до масштабування	Опис
1	Горизонтальне масштабування	Додавання нових серверів для розподілу навантаження між кількома вузлами
2	Вертикальне масштабування	Збільшення потужності існуючого обладнання (CPU, RAM)
3	Використання розподілених баз даних	Розподіл даних між кількома серверами для підвищення швидкості доступу та обробки інформації
4	Мікросервісна архітектура	Розділення системи на незалежні компоненти, що дозволяє гнучко масштабувати окремі модулі

ДОДАТОК А.7 – Ефективні UX-рішень у рекомендаційних системах

№	Елемент UX	Опис
1	Інтерактивні фільтри	Дозволяють користувачам швидко звузити вибір товарів за параметрами
2	Персоналізовані банери	Відображення спеціальних пропозицій на основі історії покупок
3	Автоматичне оновлення	Оновлення рекомендацій у реальному часі без необхідності перезавантаження сторінки
4	Зручний відгук	Легкий спосіб залишити відгук або оцінку товару

ДОДАТОК А.8 – Класифікація основних загроз безпеці у рекомендаційних системах

№	Загроза	Опис
1	Втручання у дані	Несанкціоновані зміни або видалення даних, що порушують цілісність інформації
2	Злом акаунтів	Отримання доступу до особистих облікових записів користувачів
3	DDoS-атаки	Масовані атаки, що призводять до недоступності сервісу
4	Шахрайство	Маніпуляції рейтингами та рекомендаціями за допомогою фальшивих акаунтів і ботів

ДОДАТОК А.9 – Основні методи захисту персональних даних у рекомендаційних системах

№	Методи захисту	Опис	Приклади технологій/підходів
1	Шифрування	Перетворення даних у зашифрований формат для захисту під час зберігання і передачі	AES, RSA, TLS, SSL
2	Анонімізація	Видалення або заміна ідентифікуючих ознак для збереження приватності	Маскування даних, псевдонімізація
3	Контроль доступу	Обмеження доступу до даних лише авторизованим користувачам	Ролі користувачів, ACL (Access Control Lists)
4	Багатофакторна автентифікація	Додаткові рівні перевірки особи користувача для підвищення безпеки	SMS-коди, аплети, біометрія
5	Моніторинг безпеки	Відстеження подій та спроб доступу для виявлення потенційних загроз	SIEM-системи, логування подій
6	Оновлення ПЗ	Регулярне виправлення вразливостей у програмному забезпеченні	Патчі, оновлення безпеки

ДОДАТОК А.10 – Основні методи захисту персональних даних у рекомендаційних системах

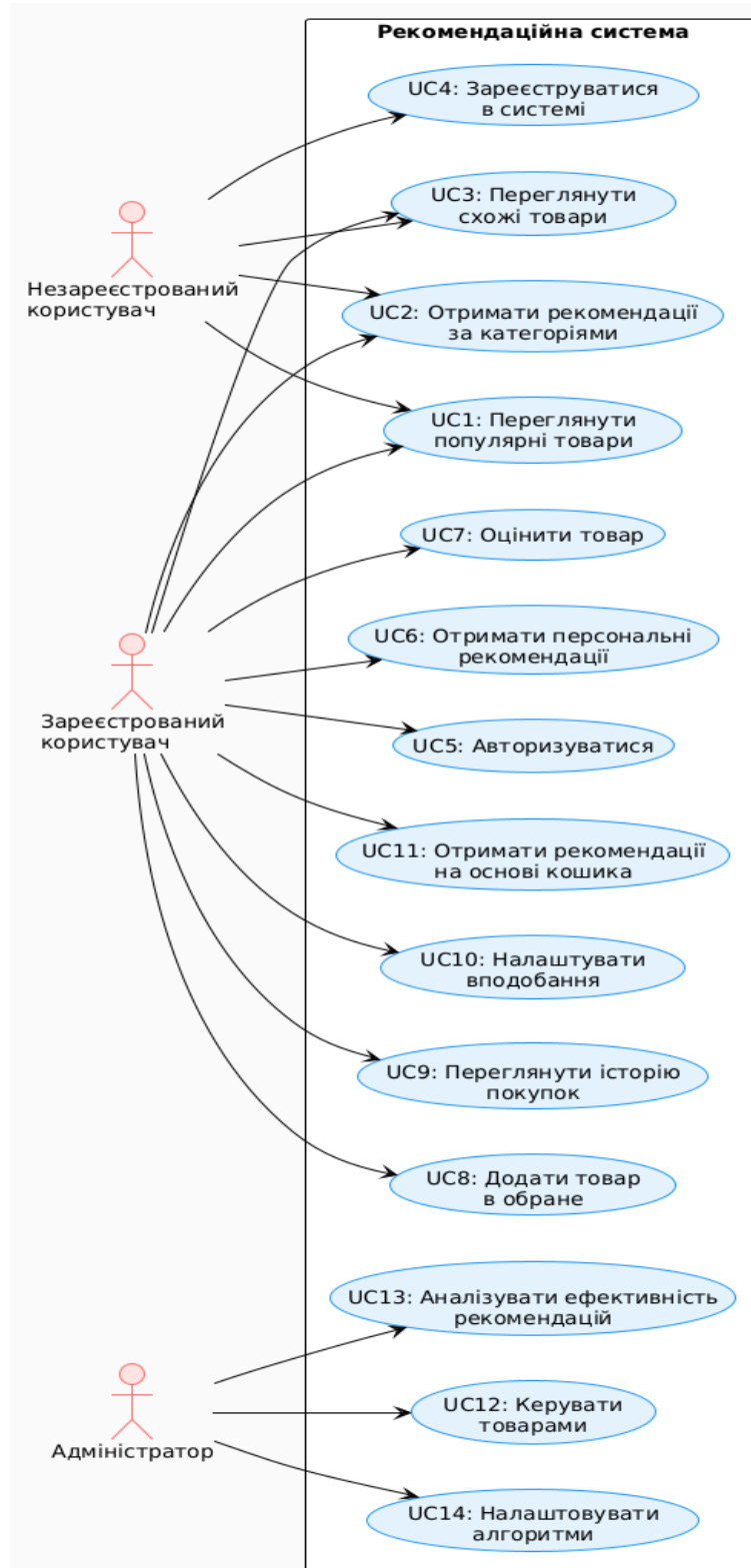
№	Методи захисту	Опис	Приклади технологій/підходів
1	Шифрування	Перетворення даних у зашифрований формат для захисту під час зберігання і передачі	AES, RSA, TLS, SSL
2	Анонімізація	Видалення або заміна ідентифікуючих ознак для збереження приватності	Маскування даних, псевдонімізація
3	Контроль доступу	Обмеження доступу до даних лише авторизованим користувачам	Ролі користувачів, ACL (Access Control Lists)
4	Багатофакторна автентифікація	Додаткові рівні перевірки особи користувача для підвищення безпеки	SMS-коди, аплети, біометрія
5	Моніторинг безпеки	Відстеження подій та спроб доступу для виявлення потенційних загроз	SIEM-системи, логування подій
6	Оновлення ПЗ	Регулярне виправлення вразливостей у програмному забезпеченні	Патчі, оновлення безпеки

ДОДАТОК А.11 – Матриця трасування функціональних вимог

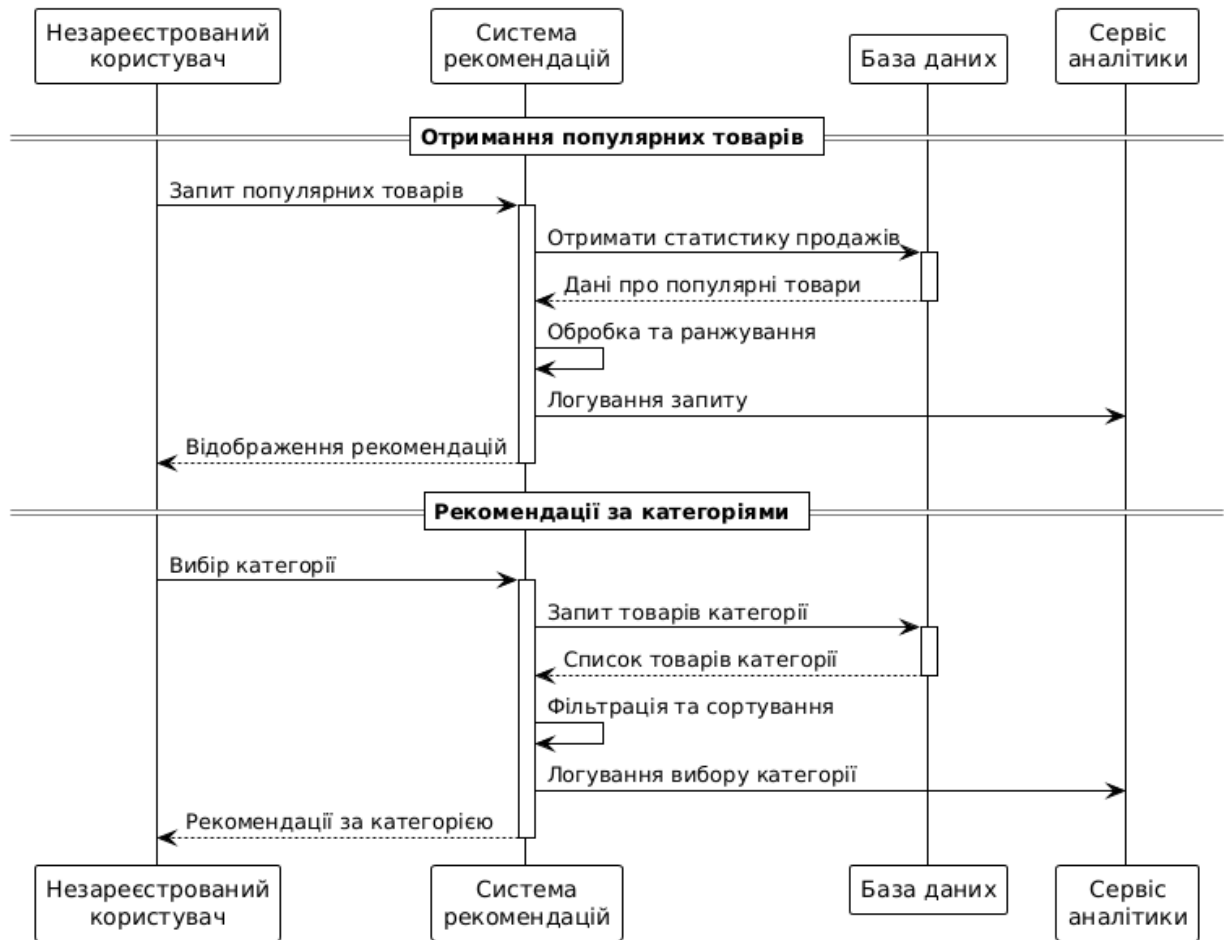
Функціональна вимога	UC1	UC2	UC6	UC11	UC13	Пріоритет
Відображення популярних товарів	✓					Високий
Категоріальні рекомендації		✓				Високий
Персоналізація			✓	✓		Критичний
Аналітика та метрики					✓	Середній
Швидкодія < 2 сек	✓	✓	✓	✓		Високий
Підтримка А/В тестування			✓		✓	Середній
Масштабованість	✓	✓	✓	✓	✓	Високий
Безпека даних			✓	✓	✓	Критичний

ДОДАТОК Б – ДІАГРАМИ

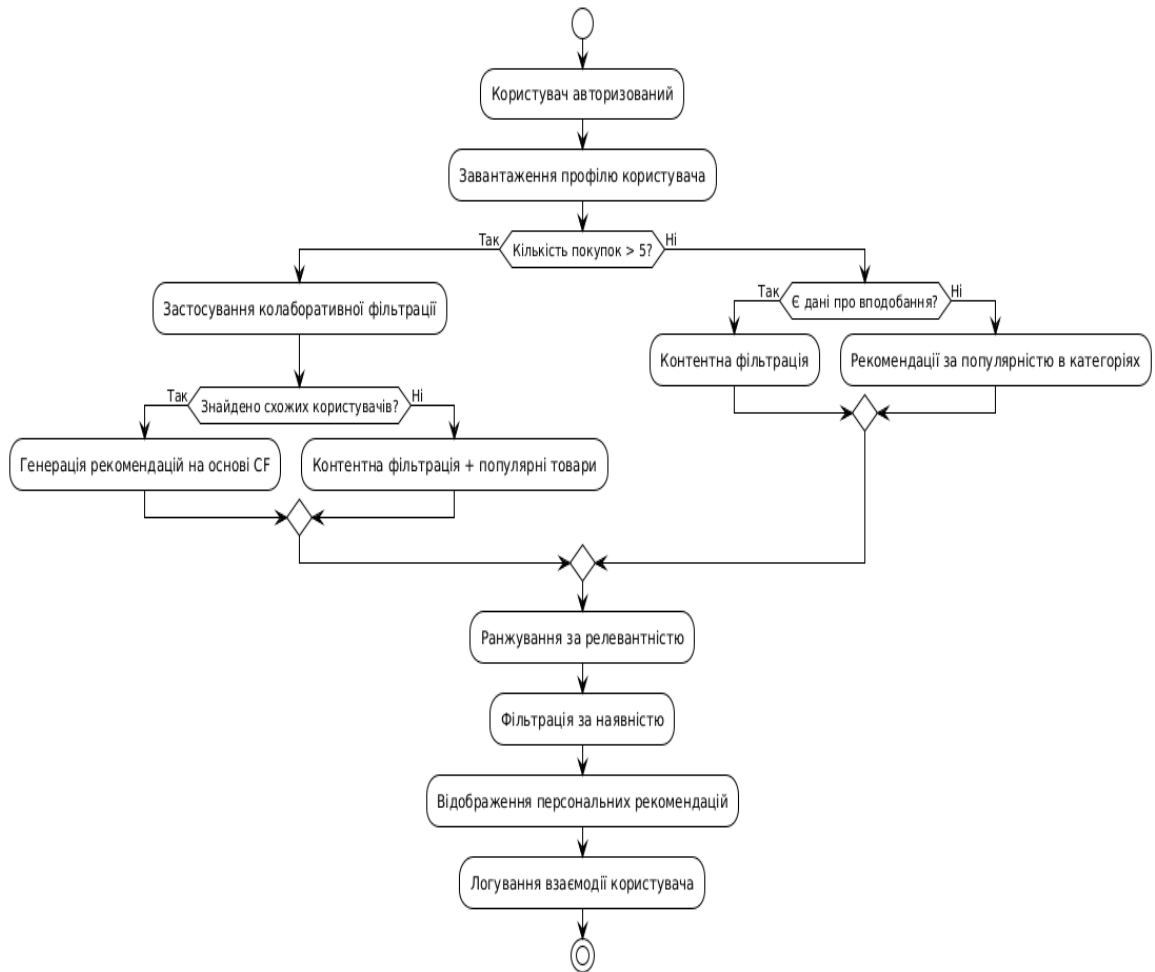
ДОДАТОК Б.1 – Загальна діаграма сценаріїв використання



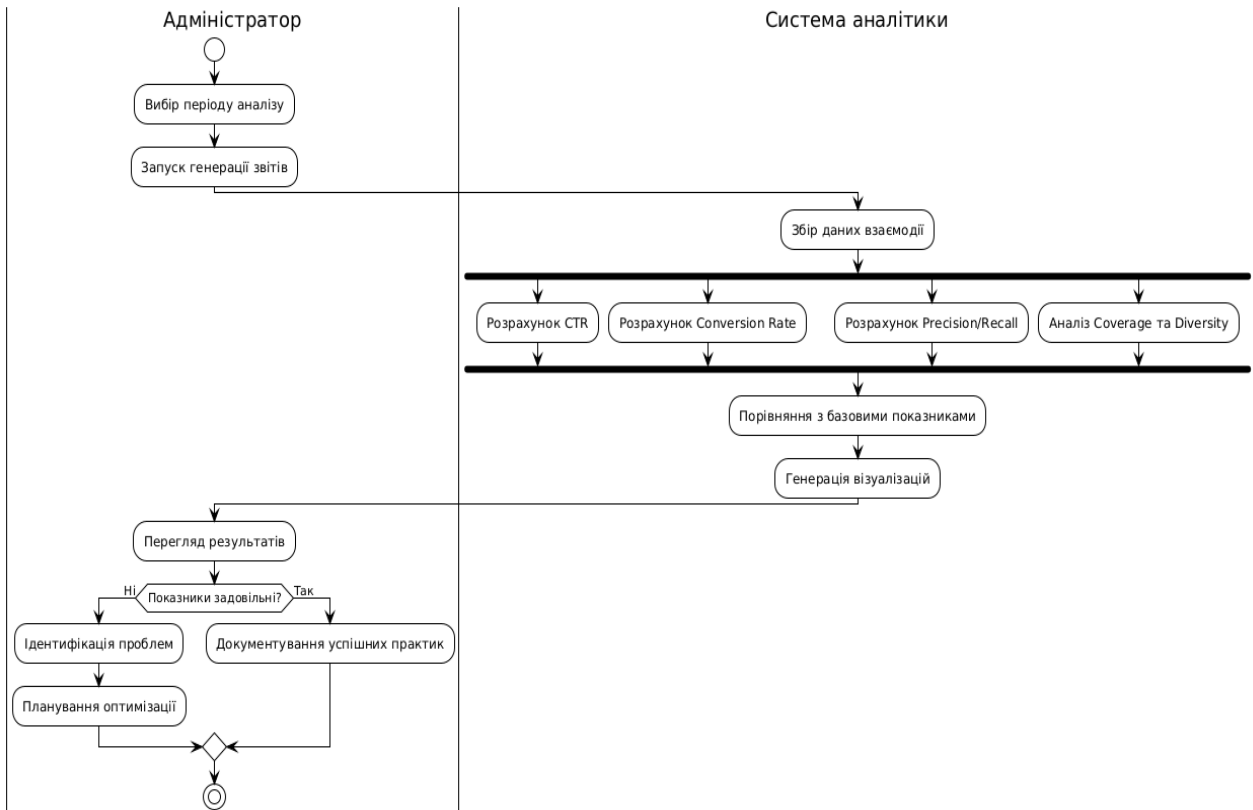
ДОДАТОК Б.2 – Діаграма послідовності для незареєстрованих користувачів



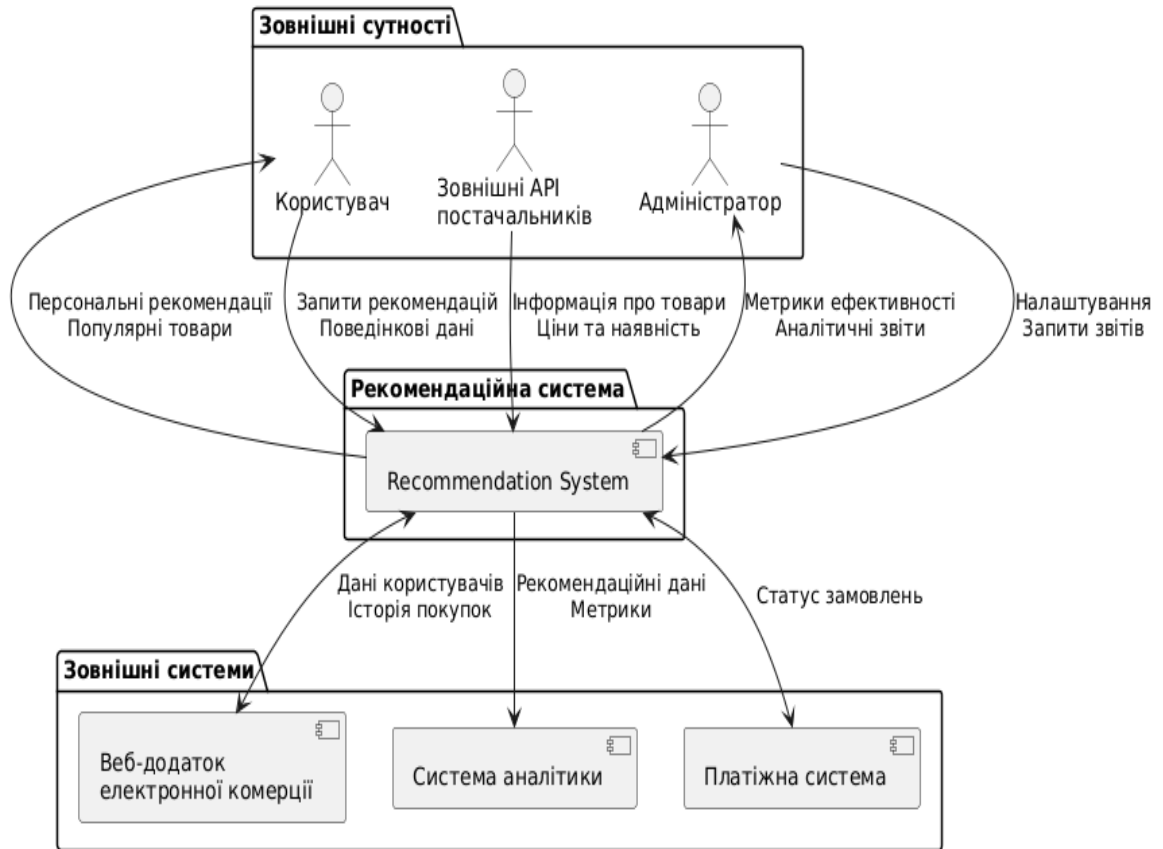
ДОДАТОК Б.3 – Блок-схема персональних рекомендацій



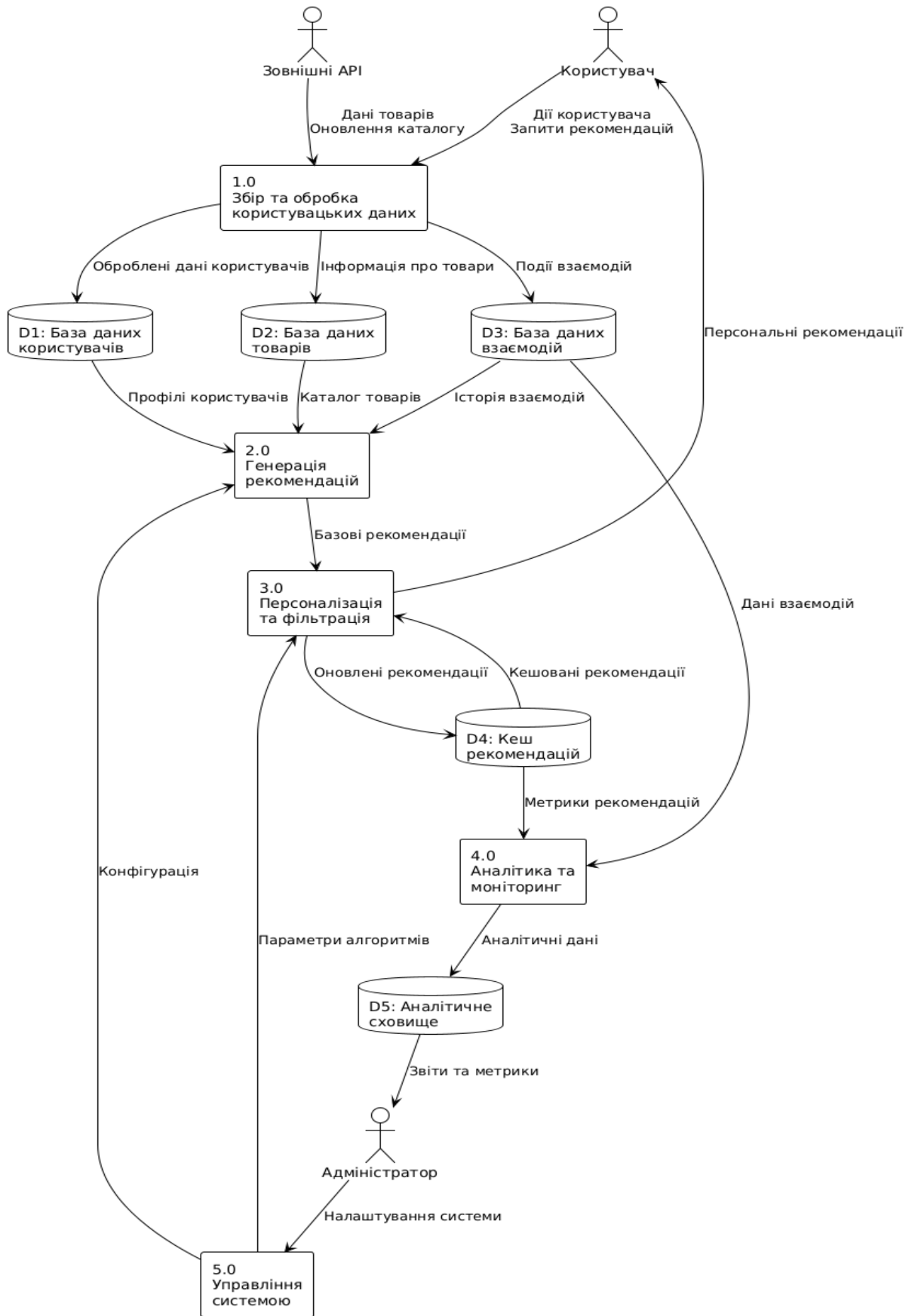
ДОДАТОК Б.4 – Діаграма активності для аналізу ефективності



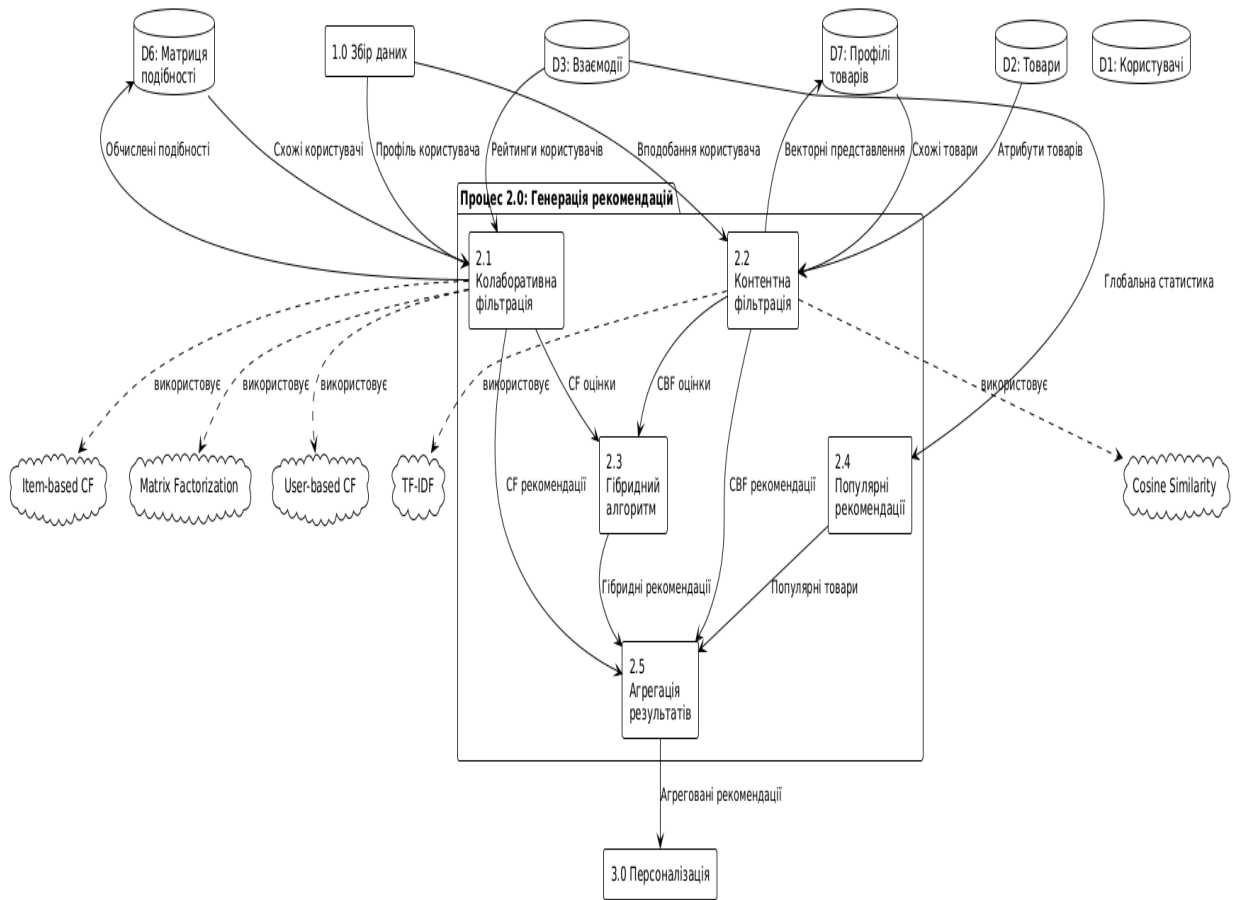
ДОДАТОК Б.5 – Контекстна діаграма рекомендаційної системи



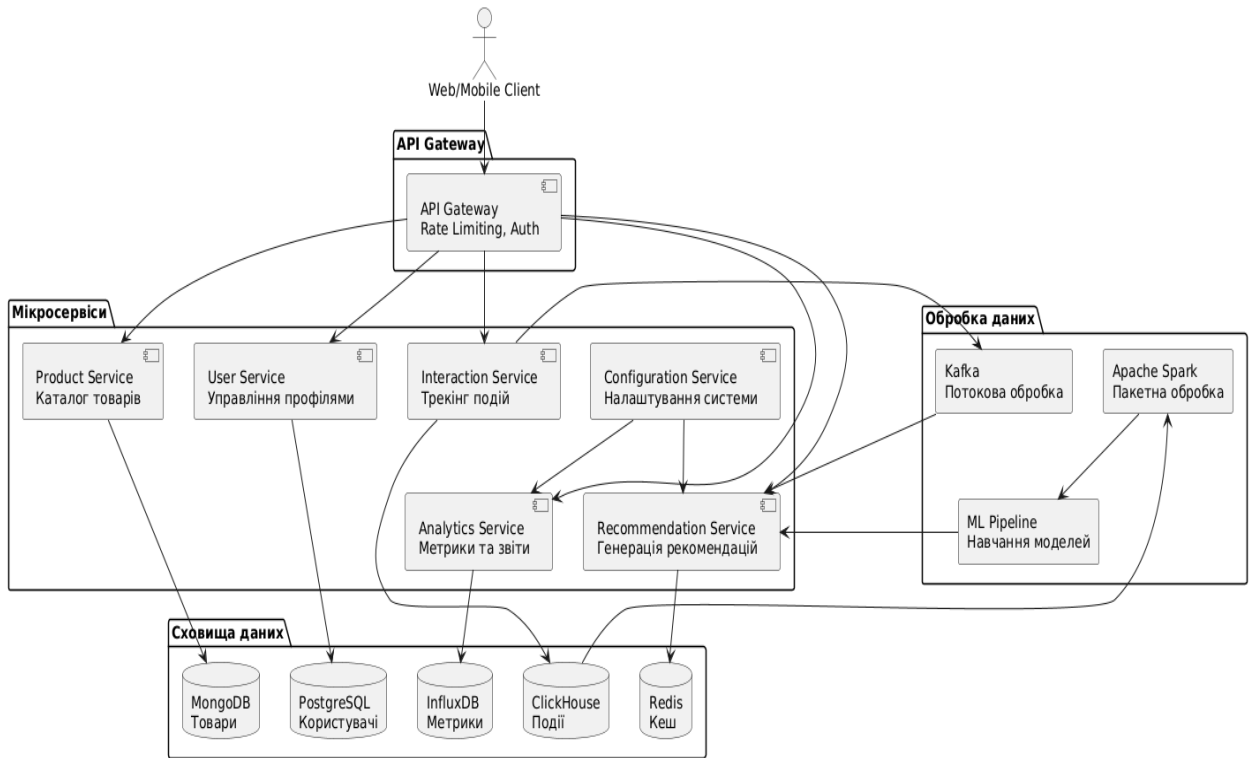
ДОДАТОК Б.6 – Діаграма нульового рівня



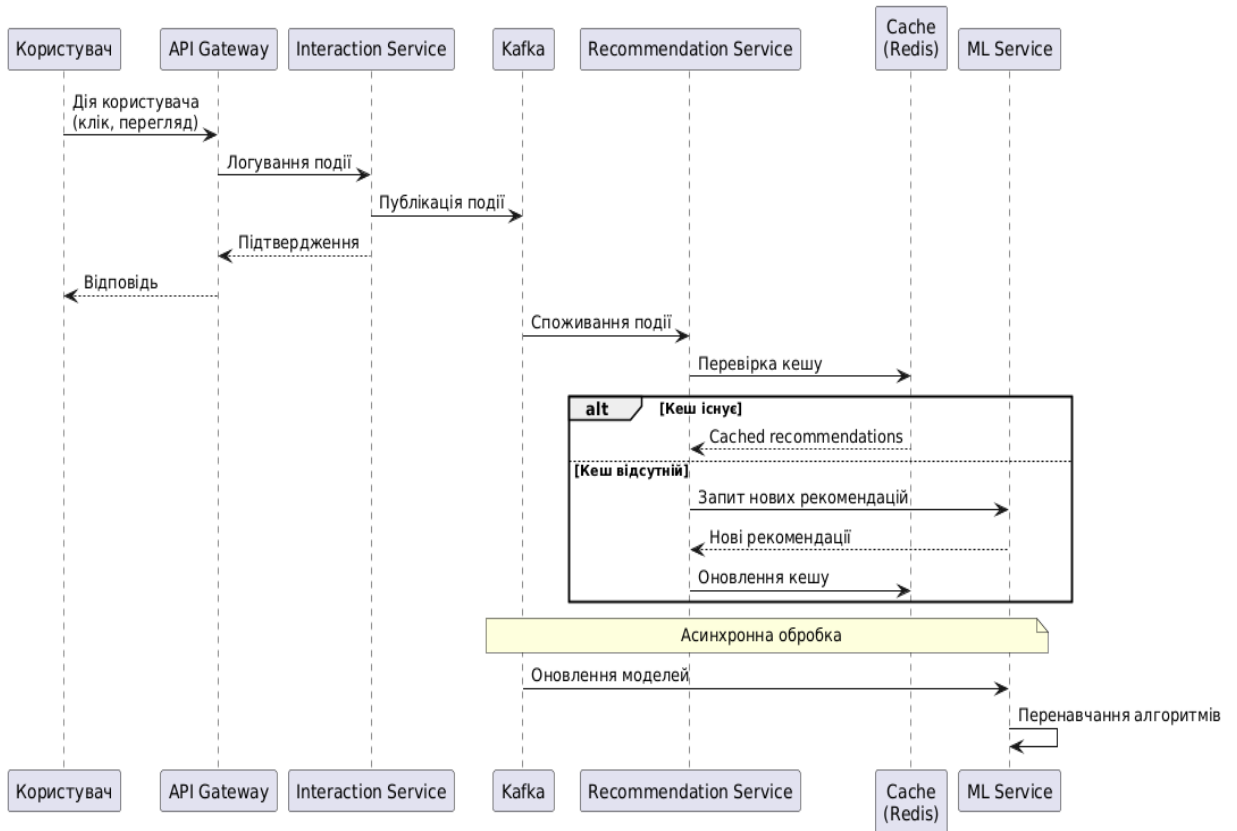
ДОДАТОК Б.7 – Діаграма першого рівня



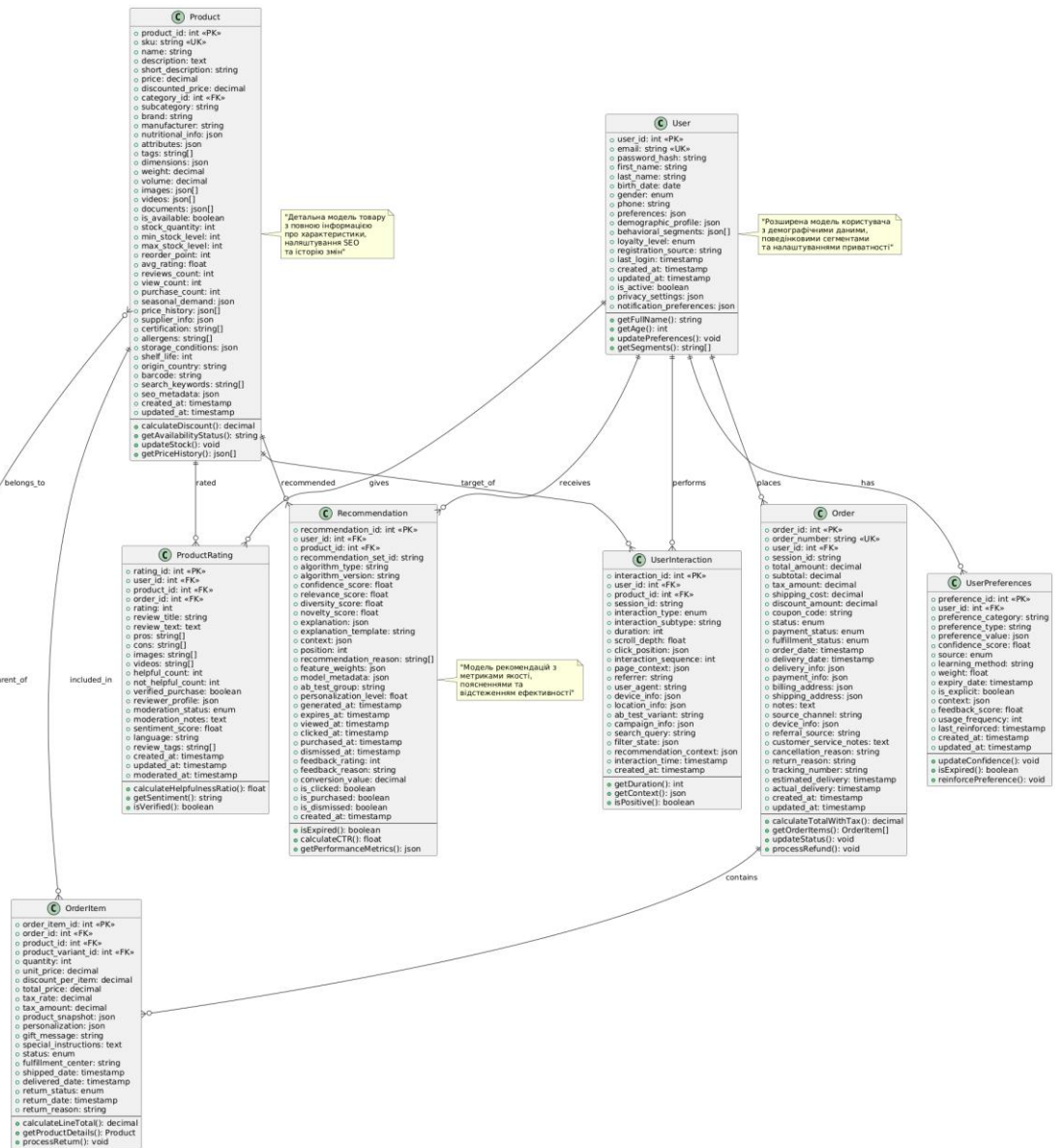
ДОДАТОК Б.8 – Діаграма архітектури рекомендаційної системи



ДОДАТОК Б.9 – Потік даних в реальному часі



ДОДАТОК Б.10 – Модель зберігання та обробки даних



ДОДАТОК Б.11 – Логічна модель даних за типами СУБД

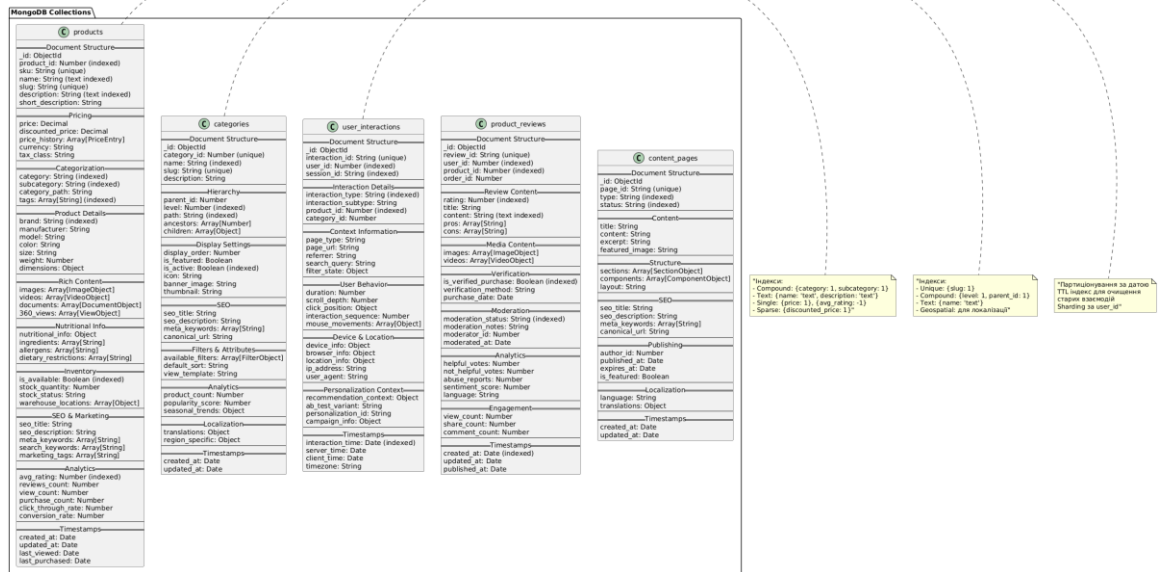
PostgreSQL Schema



«Основна таблиця користувачів з розширеними демографічними даними та налаштуваннями приватності»

«Таблиця рекомендацій з детальними метриками якості та відстеженням ефективності»

ДОДАТОК Б.12 – Основні типи даних, що зберігаються в MongoDB



ДОДАТОК В – ЛІСТИНГИ

ДОДАТОК В.1 – SQL схема для PostgreSQL

```
-- Створення типів enum
CREATE TYPE user_gender_enum AS ENUM ('male', 'female', 'other',
'prefer_not_to_say');
CREATE TYPE loyalty_enum AS ENUM ('bronze', 'silver', 'gold', 'platinum',
'diamond');
CREATE TYPE order_status_enum AS ENUM ('pending', 'confirmed',
'processing', 'shipped', 'delivered', 'cancelled', 'returned');
CREATE TYPE payment_status_enum AS ENUM ('pending', 'authorized',
'captured', 'failed', 'refunded', 'partially_refunded');
CREATE TYPE fulfillment_status_enum AS ENUM ('pending', 'picked',
'packed', 'shipped', 'delivered', 'returned');
CREATE TYPE item_status_enum AS ENUM ('pending', 'confirmed', 'picked',
'packed', 'shipped', 'delivered', 'cancelled', 'returned');
CREATE TYPE return_status_enum AS ENUM ('requested', 'approved',
'rejected', 'received', 'processed', 'refunded');
CREATE TYPE preference_source_enum AS ENUM ('explicit', 'implicit',
'inferred', 'imported');
CREATE TYPE segment_type_enum AS ENUM ('demographic', 'behavioral',
'psychographic', 'geographic', 'value_based');
-- Функції для автоматичного оновлення timestamp
CREATE OR REPLACE FUNCTION update_updated_at_column()
RETURNS TRIGGER AS $$
BEGIN
    NEW.updated_at = CURRENT_TIMESTAMP;
    RETURN NEW;
END;
$$ language 'plpgsql';
-- Тригери для автоматичного оновлення updated_at
CREATE TRIGGER update_users_updated_at BEFORE UPDATE ON users
    FOR EACH ROW EXECUTE FUNCTION update_updated_at_column();
CREATE TRIGGER update_orders_updated_at BEFORE UPDATE ON orders
    FOR EACH ROW EXECUTE FUNCTION update_updated_at_column();
-- Додаткові індекси для оптимізації запитів
CREATE INDEX CONCURRENTLY idx_orders_user_status ON orders
(user_id, status) WHERE status != 'delivered';
CREATE INDEX CONCURRENTLY idx_recommendations_active ON
recommendations (user_id, expires_at) WHERE expires_at >
CURRENT_TIMESTAMP;
```

ДОДАТОК В.2 – JavaScript код для створення індексів MongoDB

// Колекція товарів

```
db.products.createIndex({ "product_id": 1 }, { unique: true });
db.products.createIndex({ "sku": 1 }, { unique: true });
db.products.createIndex({ "category": 1, "subcategory": 1 });
db.products.createIndex({ "name": "text", "description": "text", "tags": "text" });
db.products.createIndex({ "price": 1 });
db.products.createIndex({ "avg_rating": -1 });
db.products.createIndex({ "is_available": 1, "stock_quantity": 1 });
db.products.createIndex({ "brand": 1 });
db.products.createIndex({ "created_at": 1 });
db.products.createIndex({ "tags": 1 });
```

// Колекція категорій

```
db.categories.createIndex({ "category_id": 1 }, { unique: true });
    db.categories.createIndex({ "slug": 1 }, { unique: true });
```

ДОДАТОК В.3 – Загальна діаграма сценаріїв використання

```
@startuml
!theme plain
skinparam backgroundColor #FAFAFA
skinparam actor {
    BackgroundColor #FFE4E1
    BorderColor #FF6B6B
}
skinparam usecase {
    BackgroundColor #E3F2FD
    BorderColor #2196F3
}
left to right direction

actor "Незарєєстрований\нкористувач" as Guest
actor "Зарєєстрований\нкористувач" as User
actor "Адміністратор" as Admin

rectangle "Рекомендаційна система" {
    usecase "UC1: Переглянути\нпопулярні товари" as UC1
    usecase "UC2: Отримати рекомендації\нза категоріями" as UC2
    usecase "UC3: Переглянути\нсхожі товари" as UC3
    usecase "UC4: Зарєєструватися\нв системі" as UC4
    usecase "UC5: Авторизуватися" as UC5
    usecase "UC6: Отримати персональні\нрекомендації" as UC6
    usecase "UC7: Оцінити товар" as UC7
    usecase "UC8: Додати товар\нв обране" as UC8
    usecase "UC9: Переглянути історію\нпокупок" as UC9
    usecase "UC10: Налаштувати\нвподобання" as UC10
    usecase "UC11: Отримати рекомендації\нна основі кошика" as UC11
    usecase "UC12: Керувати\нтоварами" as UC12
    usecase "UC13: Аналізувати ефективність\нрекомендацій" as UC13
    usecase "UC14: Налаштовувати\налгоритми" as UC14
}

Guest --> UC1
Guest --> UC2
Guest --> UC3
Guest --> UC4

User --> UC1
User --> UC2
User --> UC3
User --> UC5
```

User --> UC6
User --> UC7
User --> UC8
User --> UC9
User --> UC10
User --> UC11

Admin --> UC12
Admin --> UC13
Admin --> UC14

@enduml
@startuml
!theme plain
skinparam backgroundColor #FAFAFA

actor "Незареєстрований\нкористувач" as Guest

rectangle "Функції для незареєстрованих користувачів" {
 usecase "Переглянути\нпопулярні товари" as UC1
 usecase "Отримати рекомендації\нза категоріями" as UC2
 usecase "Переглянути\нсхожі товари" as UC3
 usecase "Зареєструватися\нв системі" as UC4
}

Guest --> UC1
Guest --> UC2
Guest --> UC3
Guest --> UC4

note right of UC1
Показує топ товарів
за статистикою продажів
end note

note right of UC2
Рекомендації в межах
обраної категорії
end note

note right of UC3
Товари схожі на переглянутий
за характеристиками

end note

Продовження ДОДАТКА В.3

@enduml

Діаграма послідовності для незареєстрованих користувачів

@startuml

!theme plain

participant "Незареєстрований\нкористувач" as U

participant "Система\прекомендацій" as S

participant "База даних" as DB

participant "Сервіс\паналітики" as A

== Отримання популярних товарів ==

U -> S: Запит популярних товарів

activate S

S -> DB: Отримати статистику продажів

activate DB

DB --> S: Дані про популярні товари

deactivate DB

S -> S: Обробка та ранжування

S -> A: Логування запиту

S --> U: Відображення рекомендацій

deactivate S

== Рекомендації за категоріями ==

U -> S: Вибір категорії

activate S

S -> DB: Запит товарів категорії

activate DB

DB --> S: Список товарів категорії

deactivate DB

S -> S: Фільтрація та сортування

S -> A: Логування вибору категорії

S --> U: Рекомендації за категорією

deactivate S

@enduml

Сценарії для зареєстрованих користувачів

@startuml

!theme plain

actor "Зареєстрований\нкористувач" as User

rectangle "Персональні функції" {

```
usecase "Отримати персональні\прекоме́ндації" as UC6
usecase "Оцінити товар" as UC7

usecase "Додати товар\пв обране" as UC8
usecase "Переглянути історію\ппокупок" as UC9
usecase "Налаштувати\пвподобання" as UC10
usecase "Отримати рекомендації\пна основі кошика" as UC11
}
```

```
User --> UC6
User --> UC7
User --> UC8
User --> UC9
User --> UC10
User --> UC11
```

```
UC9 ..> UC6 : <<extend>>
UC8 ..> UC11 : <<extend>>
```

```
note bottom of UC6
Використовує колаборативну
фільтрацію та історію користувача
end note
@enduml
Блок-схема персональних рекомендацій
@startuml
!theme plain
```

```
start
:Користувач авторизований;
:Завантаження профілю користувача;
if (Кількість покупок > 5?) then (Так)
  :Застосування колаборативної фільтрації;
  if (Знайдено схожих користувачів?) then (Так)
    :Генерація рекомендацій на основі CF;
  else (Ні)
    :Контентна фільтрація + популярні товари;
  endif
else (Ні)
  if (Є дані про вподобання?) then (Так)
    :Контентна фільтрація;
  else (Ні)
    :Рекомендації за популярністю в категоріях;
```

```
endif
endif
:Ранжування за релевантністю;
:Фільтрація за наявністю;
:Відображення персональних рекомендацій;
:Логування взаємодії користувача;
stop
```

```
@enduml
Адміністративні сценарії
@startuml
!theme plain
```

```
actor "Адміністратор" as Admin
```

```
rectangle "Адміністративна панель" {
    usecase "Керувати\птоварами" as UC12
    usecase "Аналізувати ефективність\прекомендацій" as UC13
    usecase "Налаштовувати\палгоритми" as UC14
}
```

```
Admin --> UC12
```

```
Admin --> UC13
```

```
Admin --> UC14
```

```
note right of UC13
    Метрики: CTR, конверсія,
    точність, покриття
end note
```

```
note right of UC14
    Налаштування параметрів
    алгоритмів ML
end note
```

```
@enduml
Діаграма активності для аналізу ефективності
@startuml
!theme plain
```

```
|Адміністратор|
start
:Вибір періоду аналізу;
:Запуск генерації звітів;
```

```
|Система аналітики|
:Збір даних взаємодії;
fork
  :Розрахунок CTR;
fork again
  :Розрахунок Conversion Rate;
fork again
  :Розрахунок Precision/Recall;
fork again
  :Аналіз Coverage та Diversity;
end fork
:Порівняння з базовими показниками;
:Генерація візуалізацій;
```

```
|Адміністратор|
:Перегляд результатів;
if (Показники задовільні?) then (Ні)
  :Ідентифікація проблем;
  :Планування оптимізації;
else (Так)
  :Документування успішних практик;
endif
stop
```

@enduml

Контекстна діаграма рекомендаційної системи

@startuml Context_Diagram

!define RECTANGLE class

```
package "Зовнішні сутності" {
  actor User as "Користувач"
  actor Admin as "Адміністратор"
  actor External as "Зовнішні API\постачальників"
}
```

```
package "Рекомендаційна система" {
  component RS as "Recommendation System"
}
```

```
package "Зовнішні системи" {
  component WebApp as "Веб-додаток\електронної комерції"
  component Analytics as "Система аналітики"
  component Payment as "Платіжна система"
}
```

User --> RS : Запити рекомендацій\nПоведінкові дані
RS --> User : Персональні рекомендації\nПопулярні товари
Admin --> RS : Налаштування\nЗапити звітів
RS --> Admin : Метрики ефективності\nАналітичні звіти
External --> RS : Інформація про товари\nЦіни та наявність
RS <--> WebApp : Дані користувачів\nІсторія покупок
RS --> Analytics : Рекомендаційні дані\nМетрики
RS <--> Payment : Статус замовлень

@enduml

Діаграма нульового рівня

@startuml

!theme plain

actor User as "Користувач"
actor Admin as "Адміністратор"
actor ExtAPI as "Зовнішні API"

rectangle "1.0\nЗбір та обробка\nкористувацьких даних" as P1
rectangle "2.0\nГенерація\nрекомендацій" as P2
rectangle "3.0\nПерсоналізація\nта фільтрація" as P3
rectangle "4.0\nАналітика та\nмоніторинг" as P4
rectangle "5.0\nУправління\nсистемою" as P5

database "D1: База даних\nкористувачів" as D1
database "D2: База даних\nтоварів" as D2
database "D3: База даних\nвзаємодій" as D3
database "D4: Кеш\nрекомендацій" as D4
database "D5: Аналітичне\nсховище" as D5

User --> P1 : Дії користувача\nЗапити рекомендацій
P1 --> D1 : Оброблені дані користувачів
P1 --> D3 : Події взаємодій

ExtAPI --> P1 : Дані товарів\nОновлення каталогу
P1 --> D2 : Інформація про товари

D1 --> P2 : Профілі користувачів
D2 --> P2 : Каталог товарів
D3 --> P2 : Історія взаємодій

P2 --> P3 : Базові рекомендації
D4 --> P3 : Кешовані рекомендації

P3 --> User : Персональні рекомендації
P3 --> D4 : Оновлені рекомендації

D3 --> P4 : Дані взаємодій
D4 --> P4 : Метрики рекомендацій
P4 --> D5 : Аналітичні дані
D5 --> Admin : Звіти та метрики

Admin --> P5 : Налаштування системи
P5 --> P2 : Конфігурація
P5 --> P3 : Параметри алгоритмів

@enduml

Діаграма першого рівня для процесу генерації рекомендацій (Level 1 DFD)

@startuml

!theme plain

rectangle "1.0 Збір даних" as P1
rectangle "3.0 Персоналізація" as P3

```
package "Процес 2.0: Генерація рекомендацій" {  
    rectangle "2.1\nКолаборативна\nфільтрація" as P21  
    rectangle "2.2\nКонтентна\nфільтрація" as P22  
    rectangle "2.3\nГібридний\nалгоритм" as P23  
    rectangle "2.4\nПопулярні\nрекомендації" as P24  
    rectangle "2.5\nАгрегація\nрезультатів" as P25  
}
```

database "D1: Користувачі" as D1
database "D2: Товари" as D2
database "D3: Взаємодії" as D3
database "D6: Матриця\nподібності" as D6
database "D7: Профілі\nтоварів" as D7

cloud "User-based CF" as ALG1
cloud "Item-based CF" as ALG2
cloud "Matrix Factorization" as ALG3
cloud "TF-IDF" as ALG4
cloud "Cosine Similarity" as ALG5

P1 --> P21 : Профіль користувача
D3 --> P21 : Рейтинги користувачів

P21 --> D6 : Обчислені подібності
D6 --> P21 : Схожі користувачі
P21 --> P25 : CF рекомендації

D2 --> P22 : Атрибути товарів
P1 --> P22 : Вподобання користувача
P22 --> D7 : Векторні представлення
D7 --> P22 : Схожі товари
P22 --> P25 : СВФ рекомендації

P21 --> P23 : CF оцінки
P22 --> P23 : СВФ оцінки
P23 --> P25 : Гібридні рекомендації

D3 --> P24 : Глобальна статистика
P24 --> P25 : Популярні товари

P25 --> P3 : Агреговані рекомендації

P21 ..> ALG1 : використовує
P21 ..> ALG2 : використовує
P21 ..> ALG3 : використовує
P22 ..> ALG4 : використовує
P22 ..> ALG5 : використовує
@enduml

Архітектура мікросервісів
@startuml Microservices_Architecture
!define MICROSERVICE component
!define DATABASE database
!define PROCESSOR rectangle

actor Client as "Web/Mobile Client"

```
package "API Gateway" {  
    MICROSERVICE GW as "API Gateway\nRate Limiting, Auth"  
}
```

```
package "Мікросервіси" {  
    MICROSERVICE US as "User Service\nУправління профілями"  
    MICROSERVICE PS as "Product Service\nКаталог товарів"  
    MICROSERVICE IS as "Interaction Service\nТрекінг подій"
```

```
MICROSERVICE RS as "Recommendation Service\nГенерація  
рекомендацій"  
MICROSERVICE AS as "Analytics Service\nМетрики та звіти"  
MICROSERVICE CS as "Configuration Service\nНалаштування  
системи"  
}
```

```
package "Сховища даних" {  
  DATABASE UserDB as "PostgreSQL\nКористувачі"  
  DATABASE ProductDB as "MongoDB\nТовари"  
  DATABASE InteractionDB as "ClickHouse\nПодії"  
  DATABASE CacheDB as "Redis\nКеш"  
  DATABASE AnalyticsDB as "InfluxDB\nМетрики"  
}
```

```
package "Обробка даних" {  
  PROCESSOR StreamProcessor as "Kafka\nПотокова обробка"  
  PROCESSOR BatchProcessor as "Apache Spark\nПакетна обробка"  
  PROCESSOR MLPipeline as "ML Pipeline\nНавчання моделей"  
}
```

```
Client --> GW  
GW --> US  
GW --> PS  
GW --> IS  
GW --> RS  
GW --> AS
```

```
US --> UserDB  
PS --> ProductDB  
IS --> InteractionDB  
RS --> CacheDB  
AS --> AnalyticsDB
```

```
IS --> StreamProcessor  
StreamProcessor --> RS  
InteractionDB --> BatchProcessor  
BatchProcessor --> MLPipeline  
MLPipeline --> RS
```

```
CS --> RS  
CS --> AS
```

```
@enduml
Підвищення конверсії через актуальні рекомендації
@startuml Real_Time_Data_Flow
participant "Користувач" as U
participant "API Gateway" as GW
participant "Interaction Service" as IS
participant "Kafka" as K
participant "Recommendation Service" as RS
participant "Cache\n(Redis)" as C
participant "ML Service" as ML

U -> GW : Дія користувача\n(клік, перегляд)
GW -> IS : Логування події
IS -> K : Публікація події
IS --> GW : Підтвердження
GW --> U : Відповідь

K -> RS : Споживання події
RS -> C : Перевірка кешу

alt Кеш існує
  C --> RS : Cached recommendations
else Кеш відсутній
  RS -> ML : Запит нових рекомендацій
  ML --> RS : Нові рекомендації
  RS -> C : Оновлення кешу
end

note over K, ML : Асинхронна обробка
K -> ML : Оновлення моделей
ML -> ML : Перенавчання алгоритмів

@enduml

Концептуальна модель даних
@startuml
!define ENTITY class
!define PRIMARY_KEY <<PK>>
!define FOREIGN_KEY <<FK>>
!define UNIQUE_KEY <<UK>>

ENTITY User {
  +user_id: int PRIMARY_KEY
  +email: string UNIQUE_KEY
```

```
+password_hash: string
+first_name: string
+last_name: string
+birth_date: date
+gender: enum
+phone: string
+preferences: json
+demographic_profile: json
+behavioral_segments: json[]
+loyalty_level: enum
+registration_source: string
+last_login: timestamp
+created_at: timestamp
+updated_at: timestamp
+is_active: boolean
+privacy_settings: json
+notification_preferences: json
--
+getFullName(): string
+getAge(): int
+updatePreferences(): void
+getSegments(): string[]
}

ENTITY Product {
+product_id: int PRIMARY_KEY
+sku: string UNIQUE_KEY
+name: string
+description: text
+short_description: string
+price: decimal
+discounted_price: decimal
+category_id: int FOREIGN_KEY
+subcategory: string
+brand: string
+manufacturer: string
+nutritional_info: json
+attributes: json
+tags: string[]
+dimensions: json
+weight: decimal
+volume: decimal
+images: json[]
```

```
+videos: json[]
+documents: json[]
+is_available: boolean
+stock_quantity: int
+min_stock_level: int
+max_stock_level: int
+reorder_point: int
+avg_rating: float
+reviews_count: int
+view_count: int
+purchase_count: int
+seasonal_demand: json
+price_history: json[]
+supplier_info: json
+certification: string[]
+allergens: string[]
+storage_conditions: json
+shelf_life: int
+origin_country: string
+barcode: string
+search_keywords: string[]
+seo_metadata: json
+created_at: timestamp
+updated_at: timestamp
--
+calculateDiscount(): decimal
+getAvailabilityStatus(): string
+updateStock(): void
+getPriceHistory(): json[]
}
```

```
ENTITY Category {
+category_id: int PRIMARY_KEY
+name: string
+slug: string UNIQUE_KEY
+description: text
+parent_id: int FOREIGN_KEY
+level: int
+path: string
+sort_order: int
+icon: string
+banner_image: string
+metadata: json
}
```

```
+is_active: boolean
+seo_title: string
+seo_description: string
+seo_keywords: string[]
+filters: json[]
+display_settings: json
+created_at: timestamp
+updated_at: timestamp
--
+getHierarchyPath(): string
+getChildren(): Category[]
+getParent(): Category
+getProductCount(): int
}
```

```
ENTITY Order {
+order_id: int PRIMARY_KEY
+order_number: string UNIQUE_KEY
+user_id: int FOREIGN_KEY
+session_id: string
+total_amount: decimal
+subtotal: decimal
+tax_amount: decimal
+shipping_cost: decimal
+discount_amount: decimal
+coupon_code: string
+status: enum
+payment_status: enum
+fulfillment_status: enum
+order_date: timestamp
+delivery_date: timestamp
+delivery_info: json
+payment_info: json
+billing_address: json
+shipping_address: json
+notes: text
+source_channel: string
+device_info: json
+referral_source: string
+customer_service_notes: text
+cancellation_reason: string
+return_reason: string
+tracking_number: string
}
```

```
+estimated_delivery: timestamp
+actual_delivery: timestamp
+created_at: timestamp
+updated_at: timestamp
--
+calculateTotalWithTax(): decimal
+getOrderItems(): OrderItem[]
+updateStatus(): void
+processRefund(): void
}

ENTITY OrderItem {
+order_item_id: int PRIMARY_KEY
+order_id: int FOREIGN_KEY
+product_id: int FOREIGN_KEY
+product_variant_id: int FOREIGN_KEY
+quantity: int
+unit_price: decimal
+discount_per_item: decimal
+total_price: decimal
+tax_rate: decimal
+tax_amount: decimal
+product_snapshot: json
+personalization: json
+gift_message: string
+special_instructions: text
+status: enum
+fulfillment_center: string
+shipped_date: timestamp
+delivered_date: timestamp
+return_status: enum
+return_date: timestamp
+return_reason: string
--
+calculateLineTotal(): decimal
+getProductDetails(): Product
+processReturn(): void
}

ENTITY UserInteraction {
+interaction_id: int PRIMARY_KEY
+user_id: int FOREIGN_KEY
+product_id: int FOREIGN_KEY
```

```
+session_id: string
+interaction_type: enum
+interaction_subtype: string
+duration: int
+scroll_depth: float
+click_position: json
+interaction_sequence: int
+page_context: json
+referrer: string
+user_agent: string
+device_info: json
+location_info: json
+ab_test_variant: string
+campaign_info: json
+search_query: string
+filter_state: json
+recommendation_context: json
+interaction_time: timestamp
+created_at: timestamp
--
+getDuration(): int
+getContext(): json
+isPositive(): boolean
}
```

```
ENTITY ProductRating {
  +rating_id: int PRIMARY_KEY
  +user_id: int FOREIGN_KEY
  +product_id: int FOREIGN_KEY
  +order_id: int FOREIGN_KEY
  +rating: int
  +review_title: string
  +review_text: text
  +pros: string[]
  +cons: string[]
  +images: string[]
  +videos: string[]
  +helpful_count: int
  +not_helpful_count: int
  +verified_purchase: boolean
  +reviewer_profile: json
  +moderation_status: enum
  +moderation_notes: text
}
```

```
+sentiment_score: float
+language: string
+review_tags: string[]
+created_at: timestamp
+updated_at: timestamp
+moderated_at: timestamp
--
+calculateHelpfulnessRatio(): float
+getSentiment(): string
+isVerified(): boolean
}
```

```
ENTITY UserPreferences {
+preference_id: int PRIMARY_KEY
+user_id: int FOREIGN_KEY
+preference_category: string
+preference_type: string
+preference_value: json
+confidence_score: float
+source: enum
+learning_method: string
+weight: float
+expiry_date: timestamp
+is_explicit: boolean
+context: json
+feedback_score: float
+usage_frequency: int
+last_reinforced: timestamp
+created_at: timestamp
+updated_at: timestamp
--
+updateConfidence(): void
+isExpired(): boolean
+reinforcePreference(): void
}
```

```
ENTITY Recommendation {
+recommendation_id: int PRIMARY_KEY
+user_id: int FOREIGN_KEY
+product_id: int FOREIGN_KEY
+recommendation_set_id: string
+algorithm_type: string
+algorithm_version: string
}
```

```
+confidence_score: float
+relevance_score: float
+diversity_score: float
+novelty_score: float
+explanation: json
+explanation_template: string
+context: json
+position: int
+recommendation_reason: string[]
+feature_weights: json
+model_metadata: json
+ab_test_group: string
+personalization_level: float
+generated_at: timestamp
+expires_at: timestamp
+viewed_at: timestamp
+clicked_at: timestamp
+purchased_at: timestamp
+dismissed_at: timestamp
+feedback_rating: int
+feedback_reason: string
+conversion_value: decimal
+is_clicked: boolean
+is_purchased: boolean
+is_dismissed: boolean
+created_at: timestamp
--
+isExpired(): boolean
+calculateCTR(): float
+getPerformanceMetrics(): json
}
```

' Relationships

```
User ||--o{ Order : "places"
```

```
User ||--o{ UserInteraction : "performs"
```

```
User ||--o{ ProductRating : "gives"
```

```
User ||--o{ UserPreferences : "has"
```

```
User ||--o{ Recommendation : "receives"
```

```
Product ||--o{ OrderItem : "included_in"
```

```
Product ||--o{ UserInteraction : "target_of"
```

```
Product ||--o{ ProductRating : "rated"
```

```
Product ||--o{ Recommendation : "recommended"
```

Product }o--|| Category : "belongs_to"

Order ||--o{ OrderItem : "contains"

Category ||--o{ Category : "parent_of"

note right of User : "Розширена модель користувача\пз демографічними даними,\пповедінковими сегментами\пта налаштуваннями приватності"

note right of Product : "Детальна модель товару\пз повною інформацією\ппро характеристики,\пналаштування SEO\пта історію змін"

note right of Recommendation : "Модель рекомендацій з\пметриками якості,\ппоясненнями та\пвідстеженням ефективності"

@enduml

Логічна модель даних @startuml

!define TABLE class

!define PRIMARY_KEY <<PK>>

!define FOREIGN_KEY <<FK>>

!define INDEX <<IDX>>

package "PostgreSQL Schema" {

TABLE users {

+user_id: SERIAL PRIMARY_KEY

+email: VARCHAR(255) UNIQUE NOT NULL

+password_hash: VARCHAR(255) NOT NULL

+first_name: VARCHAR(100)

+last_name: VARCHAR(100)

+birth_date: DATE

+gender: user_gender_enum

+phone: VARCHAR(20)

+preferences: JSONB

+demographic_profile: JSONB

+behavioral_segments: TEXT[]

+loyalty_level: loyalty_enum DEFAULT 'bronze'

+registration_source: VARCHAR(50)

+last_login: TIMESTAMP

+login_count: INTEGER DEFAULT 0

+created_at: TIMESTAMP DEFAULT CURRENT_TIMESTAMP

+updated_at: TIMESTAMP DEFAULT CURRENT_TIMESTAMP

+is_active: BOOLEAN DEFAULT TRUE

```
+privacy_settings: JSONB
+notification_preferences: JSONB
+email_verified: BOOLEAN DEFAULT FALSE
+phone_verified: BOOLEAN DEFAULT FALSE
+marketing_consent: BOOLEAN DEFAULT FALSE
+data_processing_consent: BOOLEAN DEFAULT TRUE
--
INDEX idx_users_email
INDEX idx_users_created_at
INDEX idx_users_last_login
GIN INDEX idx_users_preferences
GIN INDEX idx_users_behavioral_segments
}

TABLE orders {
+order_id: SERIAL PRIMARY_KEY
+order_number: VARCHAR(50) UNIQUE NOT NULL
+user_id: INTEGER REFERENCES users(user_id)
+session_id: VARCHAR(100)
+total_amount: DECIMAL(12,2) NOT NULL
+subtotal: DECIMAL(12,2) NOT NULL
+tax_amount: DECIMAL(10,2) DEFAULT 0
+shipping_cost: DECIMAL(10,2) DEFAULT 0
+discount_amount: DECIMAL(10,2) DEFAULT 0
+coupon_code: VARCHAR(50)
+status: order_status_enum DEFAULT 'pending'
+payment_status: payment_status_enum DEFAULT 'pending'
+fulfillment_status: fulfillment_status_enum DEFAULT 'pending'
+order_date: TIMESTAMP DEFAULT CURRENT_TIMESTAMP
+delivery_date: TIMESTAMP
+delivery_info: JSONB
+payment_info: JSONB
+billing_address: JSONB NOT NULL
+shipping_address: JSONB NOT NULL
+notes: TEXT
+source_channel: VARCHAR(50) DEFAULT 'web'
+device_info: JSONB
+referral_source: VARCHAR(100)
+customer_service_notes: TEXT
+cancellation_reason: TEXT
+return_reason: TEXT
+tracking_number: VARCHAR(100)
+estimated_delivery: TIMESTAMP
```

```
+actual_delivery: TIMESTAMP
+priority_level: INTEGER DEFAULT 1
+created_at: TIMESTAMP DEFAULT CURRENT_TIMESTAMP
+updated_at: TIMESTAMP DEFAULT CURRENT_TIMESTAMP
--
INDEX idx_orders_user_id
INDEX idx_orders_status
INDEX idx_orders_order_date
INDEX idx_orders_payment_status
GIN INDEX idx_orders_delivery_info
}

TABLE order_items {
+order_item_id: SERIAL PRIMARY_KEY
+order_id: INTEGER REFERENCES orders(order_id)
+product_id: INTEGER NOT NULL
+product_variant_id: INTEGER
+quantity: INTEGER NOT NULL CHECK (quantity > 0)
+unit_price: DECIMAL(10,2) NOT NULL
+discount_per_item: DECIMAL(10,2) DEFAULT 0
+total_price: DECIMAL(12,2) NOT NULL
+tax_rate: DECIMAL(5,4) DEFAULT 0
+tax_amount: DECIMAL(10,2) DEFAULT 0
+product_snapshot: JSONB NOT NULL
+personalization: JSONB
+gift_message: TEXT
+special_instructions: TEXT
+status: item_status_enum DEFAULT 'pending'
+fulfillment_center: VARCHAR(50)
+shipped_date: TIMESTAMP
+delivered_date: TIMESTAMP
+return_status: return_status_enum
+return_date: TIMESTAMP
+return_reason: TEXT
+replacement_item_id: INTEGER
--
INDEX idx_order_items_order_id
INDEX idx_order_items_product_id
INDEX idx_order_items_status
}

TABLE user_preferences {
+preference_id: SERIAL PRIMARY_KEY
```

```
+user_id: INTEGER REFERENCES users(user_id)
+preference_category: VARCHAR(50) NOT NULL
+preference_type: VARCHAR(100) NOT NULL
+preference_value: JSONB NOT NULL
+confidence_score: DECIMAL(3,2) DEFAULT 0.5
+source: preference_source_enum DEFAULT 'implicit'
+learning_method: VARCHAR(50)
+weight: DECIMAL(3,2) DEFAULT 1.0
+expiry_date: TIMESTAMP
+is_explicit: BOOLEAN DEFAULT FALSE
+context: JSONB
+feedback_score: DECIMAL(3,2)
+usage_frequency: INTEGER DEFAULT 1
+last_reinforced: TIMESTAMP DEFAULT CURRENT_TIMESTAMP
+created_at: TIMESTAMP DEFAULT CURRENT_TIMESTAMP
+updated_at: TIMESTAMP DEFAULT CURRENT_TIMESTAMP
--
UNIQUE INDEX idx_user_preferences_unique (user_id,
preference_category, preference_type)
INDEX idx_user_preferences_category
INDEX idx_user_preferences_confidence
GIN INDEX idx_user_preferences_value
}

TABLE recommendations {
+recommendation_id: SERIAL PRIMARY_KEY
+user_id: INTEGER REFERENCES users(user_id)
+product_id: INTEGER NOT NULL
+recommendation_set_id: VARCHAR(100) NOT NULL
+algorithm_type: VARCHAR(50) NOT NULL
+algorithm_version: VARCHAR(20) NOT NULL
+confidence_score: DECIMAL(5,4) NOT NULL
+relevance_score: DECIMAL(5,4) NOT NULL
+diversity_score: DECIMAL(5,4)
+novelty_score: DECIMAL(5,4)
+explanation: JSONB
+explanation_template: VARCHAR(100)
+context: JSONB
+position: INTEGER NOT NULL
+recommendation_reason: TEXT[]
+feature_weights: JSONB
+model_metadata: JSONB
+ab_test_group: VARCHAR(50)
```

```
+personalization_level: DECIMAL(3,2) DEFAULT 0.8
+generated_at: TIMESTAMP DEFAULT CURRENT_TIMESTAMP
+expires_at: TIMESTAMP NOT NULL
+viewed_at: TIMESTAMP
+clicked_at: TIMESTAMP
+purchased_at: TIMESTAMP
+dismissed_at: TIMESTAMP
+feedback_rating: INTEGER CHECK (feedback_rating >= 1 AND
feedback_rating <= 5)
+feedback_reason: TEXT
+conversion_value: DECIMAL(10,2)
+is_clicked: BOOLEAN DEFAULT FALSE
+is_purchased: BOOLEAN DEFAULT FALSE
+is_dismissed: BOOLEAN DEFAULT FALSE
+created_at: TIMESTAMP DEFAULT CURRENT_TIMESTAMP
--
INDEX idx_recommendations_user_id
INDEX idx_recommendations_product_id
INDEX idx_recommendations_set_id
INDEX idx_recommendations_algorithm
INDEX idx_recommendations_generated_at
INDEX idx_recommendations_expires_at
GIN INDEX idx_recommendations_context
}
```

```
TABLE user_segments {
+segment_id: SERIAL PRIMARY_KEY
+user_id: INTEGER REFERENCES users(user_id)
+segment_name: VARCHAR(100) NOT NULL
+segment_type: segment_type_enum NOT NULL
+confidence_score: DECIMAL(5,4) NOT NULL
+assignment_reason: JSONB
+segment_value: VARCHAR(100)
+priority_level: INTEGER DEFAULT 1
+effective_from: TIMESTAMP DEFAULT CURRENT_TIMESTAMP
+effective_to: TIMESTAMP
+is_active: BOOLEAN DEFAULT TRUE
+created_at: TIMESTAMP DEFAULT CURRENT_TIMESTAMP
+updated_at: TIMESTAMP DEFAULT CURRENT_TIMESTAMP
--
INDEX idx_user_segments_user_id
INDEX idx_user_segments_name
INDEX idx_user_segments_type
}
```

```
INDEX idx_user_segments_effective
}

}

note right of users : "Основна таблиця користувачів\пз розширеними
демографічними\пданими та налаштуваннями\пприватності"

note right of recommendations : "Таблиця рекомендацій з\пдетальними
метриками\пнякості та відстеженням\пефективності"

@enduml
Документні дані
@startuml
!define COLLECTION class
!define DOCUMENT class

package "MongoDB Collections" {

COLLECTION products {
  ==Document Structure==
  _id: ObjectId
  product_id: Number (indexed)
  sku: String (unique)
  name: String (text indexed)
  slug: String (unique)
  description: String (text indexed)
  short_description: String
  --
  ==Pricing==
  price: Decimal
  discounted_price: Decimal
  price_history: Array[PriceEntry]
  currency: String
  tax_class: String
  --
  ==Categorization==
  category: String (indexed)
  subcategory: String (indexed)
  category_path: String
  tags: Array[String] (indexed)
  --
  ==Product Details==
```

brand: String (indexed)
manufacturer: String
model: String
color: String
size: String
weight: Number
dimensions: Object
--
==Rich Content==
images: Array[ImageObject]
videos: Array[VideoObject]
documents: Array[DocumentObject]
360_views: Array[ViewObject]
--
==Nutritional Info==
nutritional_info: Object
ingredients: Array[String]
allergens: Array[String]
dietary_restrictions: Array[String]
--
==Inventory==
is_available: Boolean (indexed)
stock_quantity: Number
stock_status: String
warehouse_locations: Array[Object]
--
==SEO & Marketing==
seo_title: String
seo_description: String
meta_keywords: Array[String]
search_keywords: Array[String]
marketing_tags: Array[String]
--
==Analytics==
avg_rating: Number (indexed)
reviews_count: Number
view_count: Number
purchase_count: Number
click_through_rate: Number
conversion_rate: Number
--
==Timestamps==
created_at: Date

```
updated_at: Date
last_viewed: Date
last_purchased: Date
}

COLLECTION categories {
  ==Document Structure==
  _id: ObjectId
  category_id: Number (unique)
  name: String (indexed)
  slug: String (unique)
  description: String
  --
  ==Hierarchy==
  parent_id: Number
  level: Number (indexed)
  path: String (indexed)
  ancestors: Array[Number]
  children: Array[Object]
  --
  ==Display Settings==
  display_order: Number
  is_featured: Boolean
  is_active: Boolean (indexed)
  icon: String
  banner_image: String
  thumbnail: String
  --
  ==SEO==
  seo_title: String
  seo_description: String
  meta_keywords: Array[String]
  canonical_url: String
  --
  ==Filters & Attributes==
  available_filters: Array[FilterObject]
  default_sort: String
  view_template: String
  --
  ==Analytics==
  product_count: Number
  popularity_score: Number
  seasonal_trends: Object
}
```

```
--  
==Localization==  
translations: Object  
region_specific: Object  
--  
==Timestamps==  
created_at: Date  
updated_at: Date  
}  
  
COLLECTION user_interactions {  
  ==Document Structure==  
  _id: ObjectId  
  interaction_id: String (unique)  
  user_id: Number (indexed)  
  session_id: String (indexed)  
  --  
  ==Interaction Details==  
  interaction_type: String (indexed)  
  interaction_subtype: String  
  product_id: Number (indexed)  
  category_id: Number  
  --  
  ==Context Information==  
  page_type: String  
  page_url: String  
  referrer: String  
  search_query: String  
  filter_state: Object  
  --  
  ==User Behavior==  
  duration: Number  
  scroll_depth: Number  
  click_position: Object  
  interaction_sequence: Number  
  mouse_movements: Array[Object]  
  --  
  ==Device & Location==  
  device_info: Object  
  browser_info: Object  
  location_info: Object  
  ip_address: String  
  user_agent: String
```

```
--  
==Personalization Context==  
recommendation_context: Object  
ab_test_variant: String  
personalization_id: String  
campaign_info: Object  
--  
==Timestamps==  
interaction_time: Date (indexed)  
server_time: Date  
client_time: Date  
timezone: String  
}  
  
COLLECTION product_reviews {  
  ==Document Structure==  
  _id: ObjectId  
  review_id: String (unique)  
  user_id: Number (indexed)  
  product_id: Number (indexed)  
  order_id: Number  
  --  
  ==Review Content==  
  rating: Number (indexed)  
  title: String  
  content: String (text indexed)  
  pros: Array[String]  
  cons: Array[String]  
  --  
  ==Media Content==  
  images: Array[ImageObject]  
  videos: Array[VideoObject]  
  --  
  ==Verification==  
  is_verified_purchase: Boolean (indexed)  
  verification_method: String  
  purchase_date: Date  
  --  
  ==Moderation==  
  moderation_status: String (indexed)  
  moderation_notes: String  
  moderator_id: Number  
  moderated_at: Date
```

```
--  
==Analytics==  
helpful_votes: Number  
not_helpful_votes: Number  
abuse_reports: Number  
sentiment_score: Number  
language: String  
--  
==Engagement==  
view_count: Number  
share_count: Number  
comment_count: Number  
--  
==Timestamps==  
created_at: Date (indexed)  
updated_at: Date  
published_at: Date  
}  
  
COLLECTION content_pages {  
  ==Document Structure==  
  _id: ObjectId  
  page_id: String (unique)  
  type: String (indexed)  
  status: String (indexed)  
  --  
  ==Content==  
  title: String  
  content: String  
  excerpt: String  
  featured_image: String  
  --  
  ==Structure==  
  sections: Array[SectionObject]  
  components: Array[ComponentObject]  
  layout: String  
  --  
  ==SEO==  
  seo_title: String  
  seo_description: String  
  meta_keywords: Array[String]  
  canonical_url: String  
  --
```

```
==Publishing==
author_id: Number
published_at: Date
expires_at: Date
is_featured: Boolean
--
==Localization==
language: String
translations: Object
--
==Timestamps==
created_at: Date
updated_at: Date
}
```

```
}
```

' Indexes for MongoDB

note right of products : "Індекси:\n- Compound: {category: 1, subcategory: 1}\n- Text: {name: 'text', description: 'text'}\n- Single: {price: 1}, {avg_rating: -1}\n- Sparse: {discounted_price: 1}"

note right of categories : "Індекси:\n- Unique: {slug: 1}\n- Compound: {level: 1, parent_id: 1}\n- Text: {name: 'text'}\n- Geospatial: для локалізації"

note right of user_interactions : "Партиціонування за датою\nTTL індекс для очищення\nстарих взаємодій\nSharding за user_id"

@enduml