

УДК 519.854.2

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

АТЕСТАЦІЙНА РОБОТА Пояснювальна записка

Рівень вищої освіти другий (магістерський)

ГЮИК 505400.002 ПЗ

Розробка методу веб майнінга для видобутку знань про поведінку покупців
книжкового магазину
(тема)

Виконав:

Студент 2 курсу, групи СПРМ-19-1

Спеціальність 122 – Комп'ютерні науки
(код і повна назва напрямку)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне проектування

(повна назва освітньої програми)

Гуров О.О.

(прізвище, ініціали)

Керівник доц. Ситнікова П.Е.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Гребеннік І. В.

(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 – Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне проектування
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 20 ____ р.

ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

Студентові Гурову Олександр Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка метода веб майнінга для видобутку знань про поведінку покупців книжкового магазину

затверджена наказом по університету від " 02 " листопада 2020р. № 1516Ст

2. Термін подання студентом роботи 10.12.2020 р.

3. Вихідні дані до роботи Функції системи: Розробка метода веб майнінга для видобутку знань про поведінку покупців книжкового магазину. Перелік використовуваних програмних засобів: ОС Microsoft Windows 8 та вище. Технічне забезпечення: IBM-сумісний ПК з МП Pentium II та вище. Інтегроване середовище розробки intellij idea Інтернет підключення, зі встановленим браузером Google Chrome й не менш, ніж 1Гб оперативної пам'яті. Форма діалогу: веб браузер


4. Зміст пояснювальної записки (перелік питань, що потрібно розробити)

4.1 Вступ. 4.2 Аналіз проблемної області. 4.3 Розробка метода веб майнінга. 4.4 Розробка вимог до системи. 4.5 Опис прийнятих проектних рішень. 4.6 Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, плакатів)

5.1 Приклад дерева рішень (1 аркуш формату А4). **5.2** Вхідні та вихідні данні алгоритму (1 аркуш формату А4). **5.3** Множина можливого рішення (1 аркуш формату А4). **5.4** Алгоритму роботи методу (1 аркуш формату А4). **5.5** Концептуальна діаграма інформаційної системи (1 аркуш формату А4). **5.6** Діаграма декомпозиції концептуальної діаграми (1 аркуш формату А4). **5.7** Концептуальна діаграма потоків даних (1 аркуш формату А4). **5.8** Діаграма декомпозиції потоків даних (1 аркуш формату А4). **5.9** Діаграма використання системи (1 аркуш формату А4). **5.10** Діаграма послідовності дій оформлення замовлення (1 аркуш формату А4). **5.11** Карта сайту (1 аркуш формату А4). **5.12** Модель системи (1 аркуш формату А4). **5.13** Діаграма класів контролера (1 аркуш формату А4). **5.14** Інтерфейс головної сторінки (1 аркуш формату А4). **5.15** Інтерфейс головної сторінки (1 аркуш формату А4).

6. Консультанти з роботи із зазначенням розділів роботи, що їх стосуються

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		(підпис)	(дата)
<i>Розділи спеціальної частини</i>	<i>доц. Ситнікова П. Е.</i>		

7. Дата видачі завдання 10 вересня 2020 р.

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів атестаційної роботи	Термін виконання етапів роботи	Примітка
1.	<i>Отримання завдання атестаційної роботи</i>	<i>10.09.20</i>	
2.	<i>Аналіз завдання, літератури та аналогів з теми атестаційної роботи</i>	<i>21.09.20</i>	
3.	<i>Вибір засобів для розробки технічних вимог до програми</i>	<i>24.09.20</i>	
4.	<i>Структурне проектування</i>	<i>05.10.20</i>	
5.	<i>Вибір середовища розробки програми</i>	<i>11.10.20</i>	
6.	<i>Розробка метода веб майнінга</i>	<i>31.10.20</i>	
7.	<i>Розробка програми</i>	<i>07.11.20</i>	
8.	<i>Тестування програми</i>	<i>12.11.20</i>	
9.	<i>Оформлення пояснювальної записки та програмної документації</i>	<i>17.11.20</i>	
10.	<i>Оформлення графічної частини та презентаційних матеріалів комп'ютерного захисту</i>	<i>19.12.20</i>	
11.	<i>Представлення на рецензування</i>	<i>05.12.20</i>	
	<i>Представлення атестаційної роботи в ДЕК</i>	<i>05.12.20</i>	

Студент Гуров О. О.
(підпис)

Керівник роботи доцент Ситнікова П. Е.
(підпис)

РЕФЕРАТ

Пояснювальна записка до атестаційної роботи містить: 66 сторінок, 16 рисунків, 4 таблиць, 25 джерел. Графічна частина атестаційної роботи містить 16 плакатів.

Метою атестаційної роботи - розробка методу веб майнінга для пошуку знань про поведінку покупців на прикладі системи книжкового магазину.

Об'єктом дослідження – інформаційна система книжкового магазину.

Предмет дослідження – методи веб майнінга для видобутку знань.

Результат атестаційної роботи є спроектований та розроблений метод веб майнінга, впроваджений до системи книжкового інтернет магазину.

ПОШУК АСОЦІАТИВНИХ ПРАВИЛ, DATA MINING, WEB MINING, APRIORI, FP GROWTH, ECLAT, ВЕБ-ДОДАТОК, РЕКОМЕНДАЦІЙНІ СИСТЕМИ

ABSTRACT

The explanatory note to the Master Thesis contains: 66 pages, 16 figures, 4 tables, 25 sources. The graphic part of the work includes 16 posters.

The purpose of the certification work is to develop a method of web mining to find knowledge about customer behavior on the example of a bookstore system.

The object of research is the information system of a bookstore. The subject of research - web mining methods for knowledge extraction.

The result of the certification work is a designed and developed method of web mining, implemented in the bookstore online store system.

ASOCIATIVE RULES, DATA MINING, WEB MINING, APRIORI, FP GROWTH, ECLAT, WEB DATER, RECOMMENDED SYSTEMS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Аналіз електронної комерції	10
1.2 Аналіз методів добування даних	14
1.3 Аналіз моделі поведінки покупців	20
1.4 Постановка задачі на атестаційну роботу.....	21
2 РОЗРОБКА МЕТОДА ВЕБ МАЙНІНГА.....	23
2.1 Огляд методів веб майнінга.....	23
2.2 Обґрунтування вибору метода веб майнінга.....	28
2.3 Розробка метода веб майнінга.....	30
2.4 Програмна реалізація розробленого метода веб майнінга.....	35
3 РОЗРОБКА ВИМОГ ДО СИСТЕМИ	38
3.1 Визначення функціональних вимог.....	38
3.2 Розробка моделі потоків даних	41
3.3 Розробка діаграми варіантів використання	45
3.4 Розробка діаграма послідовності дій.....	47
3.5 Розробка карти сайту	49
4 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ	51
4.1 Архітектура Model View Controller	51
4.2 Обґрунтування вибору мови програмування	52
4.3 Розробка моделі	55
4.4 Розробка контролера	58
4.5 Розробка інтерфейсу	59
4.6 Впровадження розробленого метода.....	61
ВИСНОВКИ.....	62

ПЕРЕЛІК ПОСИЛАНЬ	64
ДОДАТОК А	67
ДОДАТОК Б	85
ДОДАТОК В	97

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних;

СКБД – Система керування базами даних;

CASE – Computer Aided Software Engineering (набір інструментів і методів програмної інженерії для проектування програмного забезпечення);

ERD – Entity Relationship Diagram (модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків);

MVC – Model-View-Controller (архітектурний шаблон підходу до створення об'єктно-орієнтованих web-додатків);

URL – Uniform Resource Locator (уніфікований покажчик інформаційного ресурсу).

ВСТУП

Сьогодні не можливо увити життя без використання інтернету. Інтернет охоплює багато речей та продовжує далі стрімкий розвиток. З початку це був пристрій з передачі даних. Він міг зв'язати декілька комп'ютерів або створити локальну мережу. Далі поява протоколу World Wide Web (WWW) відкрило багато можливостей не тільки дослідникам та вченим, але и звичайним людям. Веб сторінки почали стрімкий розвиток. З'являлись нові технічні рішення, поліпшувалась швидкість передачі даних. Це сприяло накопиченню великих об'ємів даних. Данні стали об'єктом дослідження і виникла така галузь, як Data Mining або добування даних.

Добування даних – це процес знаходження нових корисних знань, які знаходяться у великих масивах даних, та необхідних для пошуку практичних рішень у різних сферах.

Тому добування даних знайшло широке застосування в таких галузях як медицина або електронна комерція. Методи по видобутку даних дозволяють ефективно управляти бізнесом. Результатом практичного застосування методів на прикладі електронної комерції може бути підвищення показників продажів.

В атестаційній роботі розглядаються сучасні методи добутку знань. Після їх розглядання ми оберемо той, що найбільш відповідає нашій предметній області, розглянемо як можна модернізувати обраний метод та інтегрувати у нашу систему. Також зробимо опис предметної області та визначимо постановку завдання на дипломну роботу. Далі поставимо вимоги до розроблюваного об'єкта дослідження - книжкового інтернет магазину та на їх підставі розробимо систему та запровадимо метод веб майнінга.

Актуальність роботи полягає в тому, що вона дозволить переглянути, як на прикладі можна використовувати сучасні методи добутку знань,

інтегрувати їх до системи та модернізувати, щоб поліпшити показники продажів.

Об'єктом дослідження – інформаційна система книжкового магазину.

Предмет дослідження – методи веб майнінга для видобутку знань.

Метою атестаційної роботи є пошук знань про поведінку покупців у базі даних книжкового магазину, використовуючи методи веб майнінга.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз електронної комерції

На сьогодні використанням електронних торгових операцій є однією із ключових частин ведення успішного бізнесу. Бізнес називають такою самостійною діяльністю, яка спрямована на постійне отримання прибутку від, продажу товарів, виконання робіт, користування майном або надання різних послуг. Сучасні компанії, що мають нагоду використати інтернет, мають великі плюси перед іншими компаніями за рахунок швидкого вирішення задач, прозорого обліку документації. До електронних відносять такі середовища, у яких задіяна електронна комерція або електронний бізнес, які у свою чергу базуються на використанні цифрових технологій і тих перевагах, які вони надають. Отже, електронна комерція - це процес покупки, продажу, передачі або обміну товарами, послугами і інформацією за допомогою електронних засобів комунікації.

Таким чином, електронна комерція - це найважливіша складова частина електронного бізнесу, яка представляє собою новий спосіб організації, управління і здійснення бізнес-операцій з використанням комп'ютерів і комунікаційних мереж, тобто будь-яка форма бізнес-угоди, в якій сторони взаємодіють електронним способом, а не за допомогою фізичних операцій обміну або прямого фізичного контакту.

Електронна торгівля або електронна комерція дає можливість компаніям бути більш ефективними і гнучкими в їх внутрішньої діяльності, працювати більш тісно з їх постачальниками і оперативно реагувати на потреби і очікування клієнтів. Причому, вона дозволяє компаніям вибрати найкращих постачальників незалежно від їх географічного розташування та продавати на глобальному ринку [1].

Веб сайт з інтерактивним контентом, розташований на сервері, називають вітриною інтернет магазину. Звичайно інтернет магазини розміщують або на сервері з постійно діючим каналом зв'язку, який може бути десь далеко, або на корпоративному сервері в локальній мережі підприємства. Це обумовлено тим, що такий магазин повинен завжди мати постійний зв'язок з інформаційною системою компанії. Така система повинна забезпечувати автоматичне виконання всіх дій, які пов'язані з операціями на складі, продажем товарів, мати свій план, при виникненні надзвичайних ситуацій тощо.

У загальному вигляді мінімальні програмні та апаратні компоненти, що необхідні для роботи інтернет магазину, включають сервер та базу даних. До цього також підключені платіжні системи, або системи доставки [2].

Робота інтернет магазину атестаційної роботи буде вести свою діяльність згідно з законами України, основна мета яких отримання прибутку з продажу товарів. Усі товари нашого магазину зберігаються на складі постачальників. Попередньо керівник магазину за допомогою маркетолога обирають найкращих постачальників та укладають з ними контракт. Адміністратор заносить данні товарів до бази даних системи, також визначаючи опис та зображення книжок. Додані товари можна побачити на головній сторінці сайту. Клієнт може зайти до сайту, переглянути та замовити обрані товари, але перед цим йому потрібно зареєструватись. Нові замовлення передаються до поштової компанії, яка здійснює доставку товару. Бухгалтерія веде облік документів, у тому числі з показників продажів.

Існують такі основні типи електронної комерції: електронна торгівля, електронні гроші, електронний маркетинг, електронний банкінг та електронне страхування.

Під електронною торгівлею розуміється підприємницька діяльність з продажу товарів, виконання робіт і надання послуг, що здійснюється з

використанням інтернету. Книжковий інтернет магазин, якраз і належить до цієї галузі.

Електронну комерцію також класифікують залежно від учасників відношень. Діяльність інтернет магазину нашої атестаційної роботи тісно пов'язана з споживачами. Таке взаємовідношення називається Business-To-Consumer - бізнес для споживача, яке характеризується тим, що: компанія виробник товарів і послуг орієнтована на взаємодію з кінцевим споживачем; споживач купує товар, що задовольняє індивідуальні потреби, приймає рішення індивідуально; покупець не є експертом в товарі; цикл процесу продажу дуже короткий.

Електронний бізнес має деякі переваги:

Оскільки мережа інтернет глобальна, електронна комерція дає можливість навіть самим маленьким постачальникам здобути присутності на глобальному рівні і вести бізнес по всьому світу.

Здатність перевищити конкурентів. Електронна комерція також дає можливість постачальникам бути більш конкурентоспроможними. Щоб пропонувати розширену до і після продажну підтримку, використовують технології електронної комерції.

Повна інформованість потреб споживачів. Електронні засоби дозволяє компаніям дізнатись детальну інформацію про запити кожного окремого споживача, що у свою чергу сприяє автоматичному знаходженню товарі відповідно до вимог клієнта.

Укорочення життєвого циклу потрапляння товару до клієнтів. Електронна комерція може суттєво оптимізувати потік товарів. Товари доставляються просто від виробника до споживача, минаючи традиційні проміжні пункти у вигляді підприємств оптової та роздрібної торгівлі.

Зниження витрат. Укладення транзакції в електронному вигляді зменшує витрати на обслуговування операції на порядок, що тягне за собою нижчі ціни для споживачів [2].

Зниження фінансових витрат при запуску магазину. Так як магазин розміщує товари у інтернеті, то потрібно замовити створення веб сайту. Ціна буде залежить від того, яке наповнення сайту, складність реалізації, які технології використовувати, час розробки та загальний обсяг усіх матеріалів сайту. Але це все одно займає менше витрат, у порівнянні з відкриттям фізичного магазину. Оренда магазину може коштувати навіть більше.

Затрата часу на відкриття інтернет магазину. Сайт можна запустити протягом декількох днів після прийняття рішення про власний бізнес. Знадобиться більше часу, щоб знайти постачальників, які можуть запропонувати вигідні ціни та високоякісні товари.

Великий асортимент продукції. Якщо ви прийняли рішення співпрацювати з постачальниками, то вам не потрібно зберігати всі товари у своєму складі. Ви можете легко вказати на своєму веб сайті весь спектр товарів постачальників.

До недоліків відносять:

Проблеми ціноутворення. Знайти та порівняти ціни в інтернеті легко, тому вони знижуватимуться.

Інформаційна безпека.

Також неохопленим залишається деякий сегмент населення, що не має доступу до Інтернету [2].

Комісії платежу. Кожна платіжна система, якою користуватимуться на вашому сайті, буде стягувати свою комісію за проведені операції.

Спори з клієнтами. Клієнти можуть бути незадоволені якістю товару, його невідповідністю з картинкою на картці товару або часом доставки. При купівлі товару в звичайному магазині покупець може переконатися, що він купує саме те, що йому підходить.

Кожна сфера діяльності має свої переваги і недоліки. Але добре розуміючий на своїй системі можна знайти шляхи як повернути недоліки у переваги, або компенсувати їх чимось.

1.2 Аналіз методів добування даних

Видобуток даних - набір методів, які дозволяють знаходити у великих масивах даних нові, корисних, легких в інтерпретації знань, які потім використовують для знаходження практичних рішень у будь яких задачах. До таких відносять методи вирішення задач класифікації, прогнозування, асоціації, кластеризації, виявлення і розбір відхилень, огляд зв'язків, оцінка, візуалізація і підведення підсумків.

Усі вищезгадані завдання діляться на описові та прогнозування. Описові спрямовані на поліпшення розуміння та вираження даних що аналізувалися, головна особливість таких моделей - простота сприйняття людиною. Завдання прогнозування вирішуються в два етапи. Перш за все будується модель, завдяки даним з відомими результатами, а потім ця модель використовується для передбачення результатів на базі інших масивів даних.

Методи рішення розбиваються на дві великі групи. Перша група називається навчання з учителем, а друга навчання без учителя. Методи першої групи, що використовують учителя, працюють в два етапи. Розроблюється модель аналізованих даних, потім модель піддається навчанню. Якщо якість роботи моделі незадовільна, тоді проводять додаткове навчання. Так триває до тих пір, поки не буде досягнутий необхідний рівень якості. Навчання без учителя включає завдання, які знаходять описові моделі і закономірності.

Класифікація та регресія. У таких задачах необхідно знайти значення залежної змінної об'єкта. Такий пошук відбувається на основі значень інших змінних, які характеризують цей об'єкт. Найчастіше для вирішення задач класифікації та регресії використовують математичні функції та дерева рішень.

Умова та висновок є основою складовою правил класифікації. Умова - це перевірка усіх незалежних змінних. Висновок - це значення залежної

змінної або розподіл її ймовірності. Якщо умова - тоді висновок. Основними перевагами правил відзначають їх простоту сприйняття людиною та написання природною мовою. Легкість додавання нових правил до набору, не змінюючи існуючих.

Один із способів опису правил в ієрархічній послідовній структурі – це дерева рішень. На кожному вузлі дерева визначають певні незалежні змінні, які потім перевіряються на їх наявність. Листя дерев відповідають до класів - значенням залежної змінної. Об'єкт відносять до певного класу, якщо значення його незалежних змінних проходять перевірку на наявність певних умов, записаним у вузлах дерева.

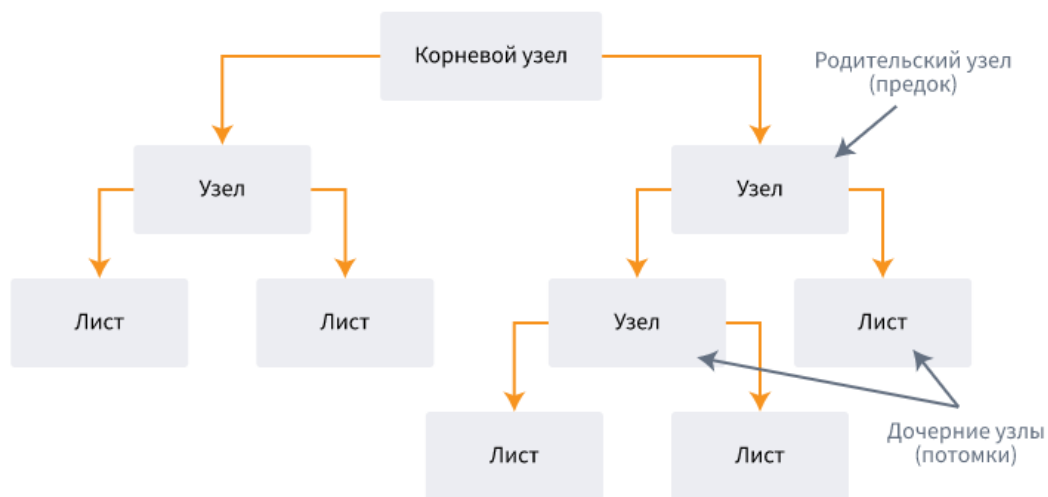


Рисунок 1.1 – Приклад дерева рішень

Пошук асоціативних правил. Асоціативні правила являють собою механізм знаходження логічних закономірностей між пов'язаними елементами. Щоб оцінити корисність та ефективність правила, використовуються різні частотні критерії, що аналізують частоту появи кандидата у масивах даних. Найголовніші з них - підтримка та достовірність.

Алгоритми пошуку асоціативних правил вибирають тих кандидатів, підтримка та достовірність яких перевищує певні заздалегідь визначені пороги. Якщо підтримка та достовірність мають велике значення, тоді

вважають що алгоритм знайшов таке правило, яке надалі можна вважати як одне із можливих рішень задачі. Заздалегідь невідомі та найцікавіші правила знаходять при маленькому значенні підтримки та достовірності. З іншого боку, маленьке значення призводить до генерування величезної кількості варіантів рішень, що у свою чергу вимагає значних обчислювальних ресурсів або до знаходження практично непридатних правил [4].

Підтримка показує, який відсоток транзакцій підтримує певне правило. Формула знаходження підтримки наведено нижче за формулою:

$$\text{Supp}_{X \Rightarrow Y} = \text{Supp}_F = \frac{|D_{F=X \cup Y}|}{|D|}.$$

Достовірністю правила X та Y вважають ставлення числа транзакцій, що містять набори X і Y , до числа транзакцій, що містять набір X . Достовірність можна розрахувати за формулою нижче:

$$\text{Conf}_{X \Rightarrow Y} = \frac{|D_{F=X \cup Y}|}{|D_X|} = \frac{\text{Supp}_{X \cup Y}}{\text{Supp}_X}.$$

Поліпшення правила є відношенням числа транзакцій, що містять набори X і Y , до добутку кількості транзакцій, що містять набір X , і кількості транзакцій, що містять набір Y . Розраховується поліпшення за формулою:

$$\text{impr}_{X \Rightarrow Y} = \frac{|D_{F=X \cup Y}|}{|D_X| \cdot |D_Y|} = \frac{\text{Supp}_{X \cup Y}}{\text{Supp}_X \cdot \text{Supp}_Y}.$$

Ці оцінки використовуються для створення правил. Під час пошуку асоціативних правил аналітик встановлює мінімальні значення цих показників, і у результаті ті правила, які не задовольняють цим умовам, відкидаються і не включаються до вирішення проблеми [5].

Алгоритм Аргіогі був розроблений ще у 1994 роках. Він використовує одне з наступних властивостей. Підтримка деякого масиву об'єктів не може бути більшою за мінімальну підтримку будь якого з його підмножин. Використання такої властивості обумовлено тим, що виявлення частих

наборів об'єктів - це операція, що генерує велику кількість обчислень, що також сприяє на часу пошуку.

Алгоритм знаходить множини, які можна часто зустріти, в кілька етапів. На кожному етапі визначаються всі елементи набори, які часто виникають. Кожен етап включає дві частин: формування кандидатів та обчислення підтримки кандидатів.

Кластеризація. Завдання кластеризації полягає в розбитті множини об'єктів на будьякі групи таким чином, щоб усередині цих груп перебували максимально близькі, схожі елементи, а елементи, що знаходяться в різних групах, максимально відрізнялися. Кластеризація має схожість із завданням класифікації в аспекті розбиття на класи, але є більш складним завданням, так як класи повинні визначитися в процесі виконання процедури, а не бути заздалегідь відомими. У деяких ситуаціях дані, якими оперують алгоритми кластеризації, містять велику кількість записів (їх кількість може досягати мільйонів), а записи, в свою чергу, містять безліч ознак (атрибутів) різних типів [6].

В основі методів кластеризації лежить порівняння об'єктів і визначення міри їх схожості. Міра подібності - величина, що має межу і відображає подібність об'єктів так, що зі збільшенням їх схожості вона зростає. Існують різні заходи близькості об'єктів, їх вибір обумовлений специфікою розв'язуваної задачі і шкалою вимірювань.

Розглянемо міри близькості, що використовують найчастіше.

Евклідова відстань використовують, щоб надати більшу вагу об'єктам, які віддалені один від одного. Нижче приведено формулу розрахунку:

$$d_2(x_i, x_j) = \sqrt{\sum_{t=1}^m (x_{it} - x_{jt})^2} .$$

Відстань Геммінга. Така відстань є звичайним середнім значенням різниці координат. Звичайно розрахунок такої відстані приводить до тих самих

результатів, що і для евклідової відстані. Відстань Хеммінга знаходиться за формулою:

$$d_H(x_i, x_j) = \sum_{t=1}^m |x_{it} - x_{jt}|.$$

Відстань Чебишова. Розраховується якщо є два об'єкта і потрібно підтвердити те, що вони не однакові, при умові що вони відрізняються від координати. Цю відстань можна дізнатися за формулою:

$$d_\infty(x_i, x_j) = \max_{1 \leq t \leq m} |x_{it} - x_{jt}|.$$

Відстань Махаланобіса. Ця міра відстані не працює добре, якщо матриця обчислюється за всією сукупністю вхідних даних. Але орієнтуючись на певну групу даних, цей показник відстані показує хороші результати. Формула Махаланобіса приведено нижче:

$$d_M(x_i, x_j) = (x_i - x_j)S^{-1}(x_i - x_j)^t.$$

Будь яку з приведених формул знаходження мір відстані можна використовувати з упевненістю, лише якщо є інформація про природу кластеризованих даних.

У результаті кластерного аналізу знаходять набір кластерів, що містить елементи вихідного набору. Кластерна модель повинна описувати кожен належність об'єктів до одного з кластерів і описувати самі кластери [5].

Основні кроки алгоритму:

- 1) Довільно обираються k початкових «центрів» кластерів.
- 2) Кожному об'єкту ставлять у відповідність кластер з найближчим «центром».
- 3) Після змін необхідно перерахувати нові «центри» кластерів.
- 4) Якщо критерій зупинки алгоритму досягнуто, - кінець, в іншому випадку - перейти до кроку 2.

Алгоритм може закінчувати роботу в наступних випадках: при мінімальному коливанні середньоквадратичної помилки або якщо другий етап не дав перехід об'єкта в інший кластер [3].

Недоліком алгоритмів k-середніх є вимога до завдання кількості кластерів, на які необхідно розбити множину.

Основною ідеєю візуального аналізу даних є подання даних у певній візуальній формі, що дозволяє аналітику переглядати дані, працювати з їх візуальним поданням, робити висновки та напряму взаємодіяти з ними. Виділяють такі основні типи методів візуалізації: геометричні перетворення, відображення іконок, орієнтовані на пікселі методи, ієрархічні образи.

Основна ідея методів геометричних перетворень - візуалізувати перетворення і проєкції даних в геометричних просторах. Підхід, заснований на відображенні іконок, передбачає кожному об'єкту даних ставити у відповідність деяку іконку, при цьому атрибути об'єкта повинні відображатися за допомогою різних візуальних властивостей іконок. Методи, орієнтовані на пікселі, використовують для подання кожного елемента даних кольорові пікселі. Ієрархічні образи використовуються для відображення ієрархій та відносин в даних.

Аналіз тексту. Інтелектуальний аналіз тексту є окремим напрямком аналізу інформації, метою якого є отримання знань та інформації з наборів текстових документів і великих обсягів тексту. Метою аналізу тональності є знаходження думок в тексті, визначення їх властивостей і емоційного забарвлення.

Аналіз на основі правил. Даний вид аналізу має на увазі наявність двох компонентів: набору правил оцінки тональності і алгоритму перевірки тексту на відповідність конкретному правилу. Як тільки ми знаходимо фрагмент тексту, відповідний якому-небудь правилом, ми можемо присвоїти йому деяке значення тональності.

Даний метод є вкрай трудомістким, тому що формування правил важко піддається формалізації і автоматизації. Також певну складність являє перевірка тексту на відповідність заданим правилам. Незважаючи на ці проблеми, даний метод аналізу широко використовується на практиці, так як дає точні результати. Точність даного підходу підвищується з ростом обсягу досліджуваних масивів тексту.

Аналіз на основі словника. За аналогією з аналізом на основі правил, аналіз на основі словника включає два основних компоненти: словник тональності і алгоритм перевірки тексту за даним словником [7].

1.3 Аналіз моделі поведінки покупців

Моделлю поведінки покупців називають таку модель, що описує реакцію клієнтів та їх характеристики при виборі певних товарів.

Вплив маркетингу на покупця можна умовно поділити на чотири складові. Це асортимент, ціна, а також стратегії розподілу та просування. Інші стимули для придбання складаються з ключових факторів навколишнього середовища покупця - економічного, науково-технічного, політичного та культурного середовища. Кожен замовник має певні характеристики, які мають велике значення, коли людина обґрунтовує свій вибір.

Характеристика покупців включає такі групи факторів: громадські, фактори культури, психологічні та особисті.

Культура є головною першопричиною, яка розкриває потреби та поведінку людини. Культура має значний вплив на поведінку споживачів.

Громадські фактори. Насамперед це соціальне становище, яке позначається належністю до соціальних класів - відносно стабільних груп у суспільстві, розташованих в ієрархічному порядку та характеризованих наявністю подібних цінностей, інтересів та поведінки серед їх членів. Люди

які належать до однакового класу поведяться приблизно так само. Соціальний клас базується на основі професії, доходу, освіти або ціннісної орієнтації [8].

Особисті фактори. Вік і стадія життєвого циклу - з віком відбуваються зміни в асортименті товарів і послуг, що купуються людьми. Певний вплив на характер товарів і послуг, придбаних людиною, робить рід її заняття. Економічне становище особистості. Люди, що належать до однієї субкультури, одного соціального класу і навіть одного і того ж роду заняття, можуть вести інший спосіб життя. Тип особистості - це сукупність психологічних характеристик людини, які забезпечують сталість її реакцій на навколишнє середовище.

Психологічні фактори. Мотив – це дуже сильна потреба, яка змушує людину шукати шляхи і засоби для її задоволення. Людина буде прагнути спочатку задовольнити найважливіші потреби, а потім наступні за значимістю. Сприйняття можна визначити як процес, за допомогою якого людина обирає, влаштовує й інтерпретує інформацію для створення картини світу навколо себе.

1.4 Постановка задачі на атестаційну роботу

Завданням атестаційної роботи є розробка метода веб майнінга для видобутку знань про поведінку покупців книжкового магазину. Для досягнення поставлено мети, можна розбити хід роботи на два етапи: розробити метод веб майнінга та розробити систему інтернет магазину з використанням метода веб майнінга.

При розробці метода треба врахувати головний критерій – швидкість та простота роботи. Обумовлено це тим, що розрахунки відбуваються на сервері, а таких запитів може бути багато, що може призвести до довгих відгуків запитів сервера.

На першому етапі треба проаналізувати методи веб майнінга та обрати такий, який найкраще б підійшов до системи та поставленої задачі. Далі потрібно розібрати обраний метод, перевірити чи є у нього недоліки. При знайденні таких, знайти рішення як модернізувати метод, що позбавитись їх. Реалізувати метод.

На другому етапі потрібно розробити систему інтернет магазину. Для цього необхідно:

- виділити функціональні вимоги;
- розібрати яким чином дані переміщуються у системі, виділити основні сутності;
- дізнатися ролі усіх користувачів системи, розподілити між ними виділені функціональні вимоги;
- побудувати діаграму послідовності дій для функції, яка використовує розроблений метод веб майнінгу, щоб детально переглянути послідовність виконання запитів;
- визначитись із структурою системи, описати обрані програмні рішення, описати обрані мови програмування;
- реалізувати систему та впровадити розроблений метод.

Отже, у результаті атестаційної роботи получимо систему книжкового інтернет магазину, яка буде використовувати метод веб майнінга, щоб дізнатися та застосувати отриманні знання про поведінку покупця.

2 РОЗРОБКА МЕТОДА ВЕБ МАЙНІНГА

2.1 Аналіз методів веб майнінга

Веб майнінг - це напрямок інтелектуального аналізу даних, у якому вирішуються завдання автоматичного добування інформації з веб-ресурсів та виявлення закономірностей з даних в інтернеті.

Іншими словами, технологія веб майнінг застосовує технологію Data Mining для аналізу неструктурованої, неоднорідною, розподіленої і значної за обсягом інформації, що міститься на веб ресурсах.

У галузі Веб майнінг виділяють наступні напрямки аналізу:

- витяг веб контенту “Web Content Mining”;
- витяг веб структур “Web Structure Mining”;
- дослідження використання веб ресурсів “Web Usage Mining”.

Витяг веб контенту. Витяг веб контенту може бути використаний при вилученні корисної інформації із вмісту веб-документів. Веб-вміст складається з декількох типів даних - тексту, зображень, аудіо, відео тощо. Дані вмісту, з яких розроблена веб-сторінка, може надати ефективні та цікаві патерни щодо потреб або поведінки користувачів. Текстові документи пов'язані з видобутком тексту, машинним навчанням та обробкою мов. Цей майнінг також відомий як видобуток тексту. Текст майнінг виконує сканування та аналіз тексту, зображень та груп веб-сторінок відповідно до вмісту введення.

Витяг веб контенту вирішує наступні задачі: знаходження структурованих даних з веб-сторінок з використанням методів машинного навчання і нейронних мереж; формування процедур уніфікації форматів представлення даних і їх інтеграції з різних джерел; виділення оцінок продуктів і послуг у відгуках, що розміщуються на форумах, в блогах і чатах.

У веб контент майнінг для кожного з трьох типів даних (структуровані, неструктуровані та квазіструктуровані) використовуються власні методи обробки, але незалежно від цього майже завжди виконується процедура перекладу даних з форми, призначеної для читання людиною, в форму, зручну для обробки комп'ютером. Така процедура називається data scraping.

Більшість методів аналізу неструктурованого тексту використовує уявлення текстового документа у вигляді множини або вектора слів. Даний підхід також широко застосовується в методах Text Mining. При цьому в такі уявлення поміщаються окремі слова без урахування їх розташування, зв'язку з іншими словами, контексту та інших лінгвістичних особливостей [5].

Витяг веб структур необхідна для виявлення інформації про структуру з інтернету. Структура веб графа складається з веб сторінок як вузлів та гіперпосилань як ребра, що з'єднують пов'язані сторінки. Структурний майнінг звичайно показує структурований звіт певного веб-сайту. Він визначає взаємозв'язок між веб-сторінками, пов'язаними інформацією. Для визначення зв'язку між двома веб-сайтами, витяг веб структур може бути дуже корисним.

Ієрархія гіперпосилань використовується ще для встановлення за допомогою пошукових машин системи гіперпосилань, які забезпечують перехід зі сторінок власного сайту до зовнішніх сайтів - наприклад, до сайтів конкурентів, де розміщена близька за змістом інформація. Далі пов'язані сторінки можна кластеризувати по логічним зв'язкам між одним або більше сайтами, які належать партнерам або конкурентам.

До результатів майнінгу веб структур можна застосувати алгоритми ранжирування, що дозволяють знайти найбільш значущі сторінки, що нагадує підрахунок індексу цитування наукових статей. Розуміння того, як контент співвідноситься з рангом сторінки, дозволяє підвищувати якість сайту.

Дослідження використання веб ресурсів необхідна для виявлення цікавих патернів з великих наборів даних. Такі патерни дозволяють зрозуміти

поведінку користувачів. Коли користувач використовує веб сторінки, данні збираються у веб логах або формах. Тому Web Usage Mining також нерідко називають лог майнінг.

Різноманітні призначені для користувача дані збираються на серверної і призначеної для користувача сторонах, а також в проксі-серверах. Веб-сервер збирає запити користувачів і зберігає їх в журналах, але достовірність даних в журналах страждає через проблеми ідентифікації сесії. Збір даних на стороні користувача здійснюється вбудованими в браузер агентами, але теоретично користувач повинен висловити свою згоду на такий спосіб відстеження. У цьому випадку проблема ідентифікації сесії виключається, проте зберігається ситуація, коли зібрані дані не цілком достовірні. Проксі-сервер отримує запити в тому ж форматі, що і веб-сервер, і може використовувати їх для узагальненої оцінки запитів.

Кластерний аналіз слугує для об'єднання об'єктів із загальними ознаками для сегментації відвідувачів сайтів і сторінок по їх відвідуваності. Це дозволяє об'єднати в групи користувачів, які мають спільні образи, - наприклад, за показниками, необхідним для цілей адресного маркетингу.

Асоціативний і кореляційний аналіз дозволяє встановлювати зв'язки між тими чи іншими явищами або процесами. Аналіз навігаційних образів служить для передбачення поведінки користувачів, що необхідно, скажімо, для розміщення реклами.

У таблиці 2.1 приведено опис категорій веб майнінга - методи, данні та їх застосування.

Таблиця 2.1 – Опис категорій веб майнінга

	Веб майнінг		
	Web Content	Web Structure	Web Usage
Тип даних	слабо структуровані	структури посилань	інтерактивні

Аналізовані дані	гіпертекстові документи	структури посилань	протоколи серверу або браузера
Представлення даних	відношення, терміни, фрази	граф	реляційні таблиці, графи
Метод	машинне навчання, асоціативні правила	власний алгоритм	статистичні, машинне навчання, асоціативні правила
Застосування	кластеризація, класифікація, пошук патернів тексту	кластеризація та класифікація	конструкція сайту, маркетинг

Веб майнінг відноситься до інтелектуального аналізу даних, але він обмежений даними, що відносяться до мережі, і виявленням закономірностей. Інтелектуальний аналіз даних - це велика концепція, яка включає в себе кілька етапів, починаючи від підготовки даних до перевірки кінцевих результатів, які призводять до процесу прийняття рішень в організації. Нижче у таблиці 2.2 наведені відмінності між веб майнінгом та дата майнінгом.

Таблиця 2.2 – Порівняння Data mining та Web mining

	Data mining	Web mining
Визначення	процес пошуку патернів та схованих знань у великих датасетів з будь-якої системи	процес пошуку патернів та схованих знань з веб ресурсів
Цільові користувачі	вчений та інженери даних.	аналітики даних

Структура	інформація з явної структури	інформація зі структурованих, неструктурованих та напівструктурованих веб-сторінок
Задачі	кластеризація, класифікація, регресія, прогнозування, оптимізація та контроль	видобуток веб-вмісту, видобуток веб-структури
Інструменти	алгоритми машинного навчання.	спеціальні інструменти для веб-майнінгу
Навички	включає підходи до очищення даних, алгоритмів машинного навчання, статистика та ймовірність	включає знання на рівні програми, інженерію даних за допомогою математичних модулів, таких як статистика та ймовірність

Оскільки в Інтернеті зараз величезна кількість інформації, користувачі можуть переглядати всі види текстових, аудіо- та відеоматеріалів. Але така різноманітність даних може містити проблеми, які можуть виникнути як при аналізі, так і при пошуку необхідної інформації в Інтернеті.

Через велику кількість різноманітних матеріалів в інтернеті, користувачеві відразу важко відшукати потрібну йому інформацію. З усіх запропонованих пошуковою системою посилань, лише деякі можуть виявитися корисними. А знайти інформацію яка непроіндексованої ще складніше.

Проблема виявлення нових знань. Витяг інформації навіть з на перший погляд потрібної інформації може виявитися досить трудомісткою справою. Також не завжди можна зрозуміти, яку саме ідею закладав автор в свої матеріали. Проблема дослідження споживачів пов'язана з наданням користувачеві інформації, яка б його зацікавила.

У веб майнінгу виділяють наступні етапи: вхідний - отримання нових даних з джерел; початкова обробка - перетворення даних до форми, необхідної для успішної побудови тієї чи іншої моделі; моделювання - побудова і застосування аналітичних моделей; аналізу - інтерпретація отриманих результатів.

Методами веб майнінга можна вирішити наступні задачі: опис відвідувачів сайту; опис відвідувачів, які здійснюють покупки в інтернет-магазині; визначення типових сесій і навігаційних шляхів користувачів сайту; визначення груп або сегментів відвідувачів; знаходження залежностей при користуванні послугами сайту.

2.2 Обґрунтування вибору метода добування даних

Мета атестаційної роботи – розробити метод веб майнінгу. Під час аналізу предметної області ми змогли з'ясувати, що таке добування даних, веб майнінг, дізналися про задачі, які вони вирішують та про алгоритми що використовуються. Однією з особливостей нашого алгоритму – він повинен знаходити знання про поведінку покупців. Але покупців не звичайного магазину, а електронної комерції - інтернет магазину книжок. Як ми дізналися раніше, електронна комерція інтернет магазинів використовують веб сторінки. В свою чергу сторінки в інтернеті пов'язані з веб майнінгом. Отже якщо все підсумувати, аналіз ринкового кошика дуже добре відповідає до усіх особливостей.

Аналіз ринкового кошика - процес пошуку найбільш типових шаблонів покупок в супермаркетах. Він проводиться шляхом аналізу баз даних транзакцій з метою визначення комбінацій товарів, пов'язаних між собою. Іншими словами, камера визначатиме товарів, наявність яких в транзакції впливає на ймовірність появи інших товарів або їх комбінацій.

Результати, отримані за допомогою аналізу ринкового кошика, дозволяють оптимізувати асортимент товарів і запаси, розміщення їх у торговельних залах, збільшувати обсяги продажів за рахунок пропозиції клієнтам супутніх товарів [9].

Для ефективної роботи інтернет-магазинів недостатньо просто виставити товари на сайті, зробити красивий інтерфейс або додати детальний опис. Впровадження додаткових аналітичних компонентів дозволило б отримати більш детальну інформацію про уподобання або залежності, які могли б виникнути в ході використання сайту клієнтами [10].

Інфокомунікації передбачають роботу з інформаційними об'єктами, обробку та аналіз даних. Під терміном «Управління даними» загалом розуміють комплекс дій, спрямованих на поліпшення якості корпоративних даних. Одним із типових процесів управління даними є інтелектуальний аналіз даних, спрямований на отримання корисної інформації та прийняття рішень на основі цієї інформації (видобуток даних та виявлення знань). Концепція видобутку великих даних є основним етапом у процесі виявлення знань у даних, тобто пошуку залежностей, закономірностей та відносин, що базується на певних формах повторення комбінацій даних у базах даних. Операції цього типу включають пошук та побудову цікавих залежностей, їх оцінку та фільтрацію. В даний час, коли апаратні та програмні технології для роботи з великими даними широко розповсюджені та використовуються, виявлення прихованих та не очевидних залежностей у даних стає завданням найвищого пріоритету.

Виявлення правил асоціації у великих даних спрямоване на пошук закономірностей (логічних залежностей) у базах даних у вигляді логічних правил. Кожне правило повинно мати міру правильності та рішучості, виражену цифровим способом. Такі числові показники не повинні бути нижчими за деякі попередньо визначені пороги [11].

Для вирішення завдання аналізу ринкової кошика використовуються асоціативні правила. Асоціативні правила гнучкі та дозволяють знаходити стійкі комбінації товарів, які дуже корисними для збільшення продажів. Але головний недолік таких алгоритмів – це великий обсяг розрахунків. Для кожної комбінації потрібно розрахувати основні показники, які відповідають про те, наскільки ці комбінації стійкі та якісні. Звичайно використовують підтримку, достовірність та поліпшення. І чим більше товарів, тим більше комбінацій і розрахунків. Це також і стосується алгоритма Apriori. Такі метод будуть довго опрацьовуватись на сервері, та буде навантажувати систему, що може призвести до збоїв у роботі чи довгого відгуку запитів. Тому нам потрібно буде самим розробити такий алгоритм, який зміг б використовуючи методи асоціативних правил, знаходити рішення, не використовуючи багато ресурсів сервера.

2.3 Розробка метода веб майнінга

Отже головною відмінністю нашого алгоритму повинна бути його оптимальність. Найбільш популярний алгоритм пошуку асоціативних правил – це алгоритм Apriori. Властивість анти-монотонності головна його особливість. Такий підхід вирішення задачі спрощує пошук, але навіть цього не достатньо для ефективної роботи системи. Базується цей алгоритм на тому, що підтримка двох елементів завжди буде однаковою або більшою, якщо додати третій елемент. Завдяки цьому ми зможемо зменшити розмір простору пошуку. Але треба знати, яке значення підтримки буде “граничною”. При одному і тому же значенні, на різних комбінаціях вхідних даних, воно може бути як і відповідним або взагалі руйнувати і відхиляти забагато елементів.

Наступна відмінність – нам не потрібно знаходити комбінацій, які б давали нам декілька потрібних елементів. Результат роботи алгоритму повинен бути лише один найбільш придатний елемент. І це тому, що ми

використовуємо асоціативні правила для аналізу ринкового кошику, щоб потім пропонувати клієнту купити ще якийсь товар. Спрямувати увагу саме на одному товарі. Алгоритм зможе знайти комбінацію з безліч товарами, які в результаті будуть не потрібні.

Отже ми змогли з'ясувати дві особливості алгоритму, по перше це оптимальність і по друге це результат лише один елемент. Спробуємо уявити роботу такого методу, як чорний ящик. На вході до алгоритму ми повинні передати якусь інформацію, яка послугує для знаходження оптимального рішення. Звичайно на вході ми передаємо інформацію про те, які товари (елементи) знаходяться у кошику клієнта. Також нам необхідно знати а які взагалі товари існують і інформацію про транзакції з цими ж товарами. Інформацію про товари ми зможемо вилучити з транзакцій – це буде унікальний набір усіх елементів, які існують у транзакціях. Ще нам потрібні деякі константи критерії. Наприклад в алгоритмі Аргіогі – це так звані граничні значення підтримки, достовірності та поліпшення, для зменшення простору пошуку. Але як ми зазначили раніше, підібрати оптимальні значення важко, і було б добре взагалі не використовувати їх. Така гнучкість може бути корисна, коли ми самі аналізуємо якісь дані і можемо власноруч задавати та дивитись на результат. Але у нашому випадку усі розрахунки мають бути автоматичними. Таким чином ми визначили, що вхідні данні – це комбінація товарів з кошика клієнта та транзакції системи. Графічно це представлено нижче на зображенні.

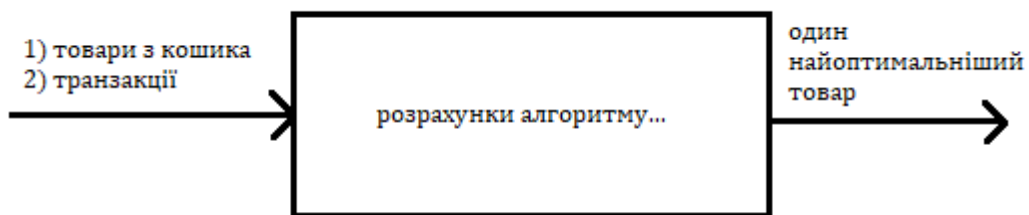


Рисунок 2.1 – Вхідні та вихідні данні алгоритму

Наступне з'ясуємо де взагалі знаходиться рішення. Якщо множина A – множина усіх елементів системи, множина B – елементи кошика клієнта, тоді рішення буде знаходитися десь у множині C , яке у свою чергу буде різницею двох множин $A - B$. Графічно це представлено на Рис. 2.2. Кожен елемент з множини C може стати рішенням, тому прозвемо її кандидатами.

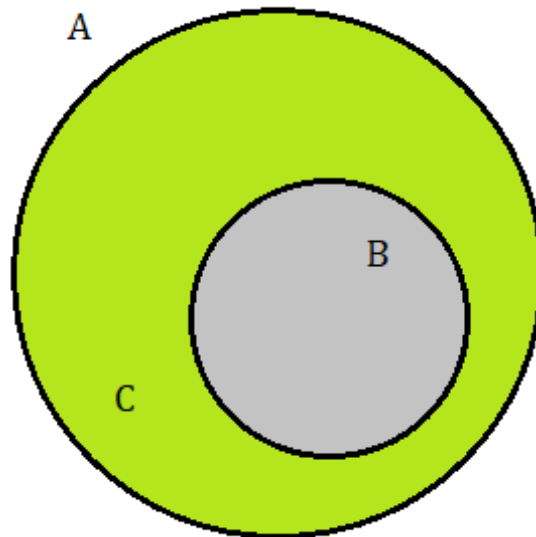


Рисунок 2.2 – Множина можливого рішення

Що стосується до показників якості кандидатів, будемо розраховувати значення підтримки, достовірності та поліпшення. Усі формули розглянуто при аналізі предметної області. Спробуємо підійти до рішення з трохи іншого боку. Поки алгоритм Аргіогі перебирає усі можливі комбінації у пошуку оптимальних комбінацій, ми ж уявимо що оптимальне рішення вже знаходиться у множині C , тоді нам залишилось з'ясувати, а яких з них той самий. Але як тоді розраховувати значення якості показників асоціативних правил. Ітеративно перебирати усі можливі комбінації кандидатів не є ефективним. Нижче приведена формула розрахунків підтримки.

$$\text{Supp}_{X \Rightarrow Y} = \text{Supp}_F = \frac{|D_{F=X \cup Y}|}{|D|}$$

Як ми бачимо, це якість відношення між числом транзакцій, у яких з'являються наслідок Y , при умові X , до загальної кількості транзакцій. Але чітка умова X веде до ітеративного рішення. Таким чином необхідно розрахувати усі можливі комбінації, при яких з'являється кандидат та товари з кошика клієнта. А як ми зазначали у відмінності алгоритму – багато розрахунків буде сильно навантажувати систему. Тому на цьому етапі будемо розраховувати показники якості кандидата інакше. Наша умова X не буде кількістю усіх транзакцій конкретної комбінації елементів, а кількістю усіх транзакцій, у яких є хоча б один елемент з кошика клієнта. Тобто D_F – це кількість таких транзакцій, у котрих присутній кандидат та хоча б один елемент з кошика клієнта. Таким чином ми уникаємо від ітеративності, і за один раз знаходимо значення показника якості кандидата. Так само будемо розраховувати значення достовірності та поліпшення.

Тоді ми дізнаємося значення трьох показників якості кожного з кандидатів. Але потрібно якось звести три показника до одного. Значення підтримки та достовірності у інтервалі від 0 до 1 і чим ближче до 1 тим правило якісніше. Поліпшення приймає значення від 0 до нескінченності. Значення менші 1 вказують на те, що умова і наслідок зустрічаються в транзакціях частіше окремо, ніж разом. Правило, що має значення більше 1, може вважатися якісним. Щоб дізнатись про якість єдине значення якості, необхідно знайти їх суму, але насамперед підтримка має бути нормалізована. Тоді треба знайти максимальне та мінімальне значення та уявити результати також у інтервалі від 0 до 1. Таким чином, кожен кандидат матиме значення якості, у інтервалі від 0 до 3, і чим більше значення, тим кандидат кращий.

Можна виділити три основних етапів, спочатку визначення кандидатів, потім проведення розрахунків їх якості і завершальний етап – знаходження найкращого. Графічно етапи можна представити на Рис. 2.3. Таким чином ми відійшли від абстрактного уявлення роботи чорного ящика, визначивши етапи пошуку рішення.

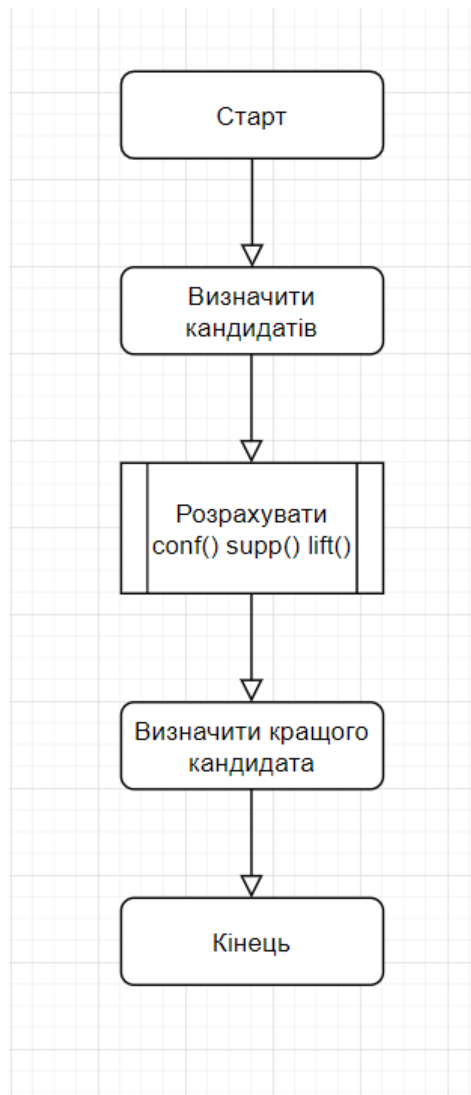


Рисунок 2.3 – Алгоритму роботи методу

Тепер підсумуємо інформацію та опишемо чітку роботу алгоритму крок за кроком. Будемо дотримуватися вище зазначеного поділу на три основні етапи.

1) Визначення кандидатів:

- знаходження множини кандидатів як різниця усіх унікальних товарів та товарів з кошику клієнта.

2) Розрахунок показників якості кожного кандидата:

- розрахунок підтримки кандидатів;
- розрахунок достовірності кандидатів;
- розрахунок поліпшення кандидатів.

3) Визначення кращого кандидата:

- приведення поліпшення до нормалізованого вигляду та сумування усіх показників якості кожного кандидата;
- знаходження кандидата з найбільшим показником якості.

2.4 Програмна реалізація розробленого метода веб майнінга

Тепер, коли ми маємо чітку логіку алгоритму, ми можемо приступити до написання коду. Так як серверна частина у нас буде використовувати мову програмування JavaScript, то і алгоритм буде написаний на ньому. Основні підстави вибору мови програмування буде розглянуто у наступних розділах. Нижче приведено код розробленого метода.

Лістинг 2.1 – Код розробленого методу

```
function calculate(transactions, userBasket) {
  let maxLift = 0;
  let minLift = Number.MAX_SAFE_INTEGER;
  const candidates = [...new Set(transactions.flat())]
    .filter(candidate => !userBasket.includes(candidate))
    .reduce((result, candidate) => ({
      ...result,
      [candidate]: {},
    }), {});
  Object.keys(candidates).forEach(candidate => {
    const suppXY = transactions
      .filter(transaction => (
        transaction.includes(candidate)
        && transaction.some(transactionItem =>
userBasket.includes(transactionItem)))
      ).length / transactions.length;
    const suppY = transactions
      .filter(transaction =>
userBasket.includes(transactionItem))
      .length / transactions.length;
    const suppX = transactions
      .filter(transaction => transaction.some(transactionItem =>
userBasket.includes(transactionItem)))
      .length / transactions.length;
    const lift = suppXY / (suppX * suppY);
    candidates[candidate].supp = suppXY;
    candidates[candidate].conf = suppXY / suppX;
  });
}
```

```

    candidates[candidate].lift = lift;
    if (lift > maxLift) {
      maxLift = lift;
    }
    if (lift < minLift) {
      minLift = lift;
    }
  });
  Object.entries(candidates)
    .forEach(([candidate, candidateData]) => {
      const lift = (candidateData.lift - minLift) / (maxLift -
minLift);
      candidates[candidate].lift = lift;
      candidates[candidate].quality = lift + candidateData.supp +
candidateData.conf;
    });
  return Object.entries(candidates).find(([key, value]) =>
value.quality === Math.max(...Object.values(candidates).map(candidate
=> candidate.quality)))[0];
}

```

Усі зазначені кроки реалізовані. У результаті метод повертає інформацію найкращого кандидата, а саме його назву та id. Щоб його використати достатньо підключити його та викликати.

Щоб дізнатися наскільки розроблений метод є ефективним, проведемо його тестування. Для цього згенеруємо данні транзакцій та кошика клієнта. Транзакції можуть містити до 20 товарів. Проведемо розрахунки при різних кількостях вхідних транзакцій. Результати проведення тестування приведено у Таблиці 2.3.

Таблиця 2.3 – Результати проведення тестування методу

Число транзакцій	Час пошуку
1 000	0.013 - 0.019
4 000	0.029 - 0.038
10 000	0.072 – 0.081
20 000	0.128 – 0.137
50 000	0.384 - 0.328

Якщо приблизно порівняти результати нашого методу та наприклад алгоритму Аргіогі, то наш буде у кілька десятків разів скоріше. Основні вимоги до алгоритму виконані, наступний кроком буде реалізація системи та впровадження розробленого метода до неї.

3 РОЗРОБКА ВИМОГ ДО СИСТЕМИ

3.1 Визначення функціональних вимог

Для того, щоб визначити функціональні вимоги, розробимо діаграми з використанням стандарту IDEF0 - методологія функціонального моделювання, призначена для формалізації і опису бізнес-процесів.

Для опису взаємодії системи з навколишнім середовищем, розробляють контекстну діаграму. На такій діаграмі розташовують посередні єдиний блок, який є представленням усієї системи та стрілки з чотирьох сторін. Призначення стрілок залежать з якого саме боку вони розташовані. Стрілки зліва значать вхідну інформацію, яка потім обробляється системою, результати яких розташовують стрілками з правого боку. Зверху стрілки описують інформацію, яка буде потрібна для управління функціональними процесами системи, а знизу стрілки називають механізми, які показують хто або за допомогою чого функція виконується. Контекстна діаграма є обов'язковою і позначають її назвою А-0. Важливо правильно підібрати назву такого блоку, тому що це буде загальним його ім'ям для нашої системи. Стрілки також мають своє ім'я. Контекстна діаграма дозволяє побачити область моделювання системи та побачити її межі.

Після розробки концептуальної діаграми, переходять до розробки діаграм декомпозицій. Процес розбиття функціонального блоку на окремі його фрагменти, та відображення їх взаємодії називають декомпозицією. Процес декомпозиції триває до тих пір, поки систему можна розбити на більш дрібні функціональні блоки. Це необхідно для того, щоб детально розглянути функції системи, та то, як вони взаємодіють між собою. Після кожної декомпозиції проводиться її аналіз на наявність невідповідностей або помилок. Після їх виправлення переходять до наступної декомпозиції. Таким чином досягається точність опису системи.

На рис. 3.1 представлена концептуальна діаграма з назвою управління продажу книжок. Розглядається система з погляду її адміністратора. Мета – розробити функціональні вимоги до системи.

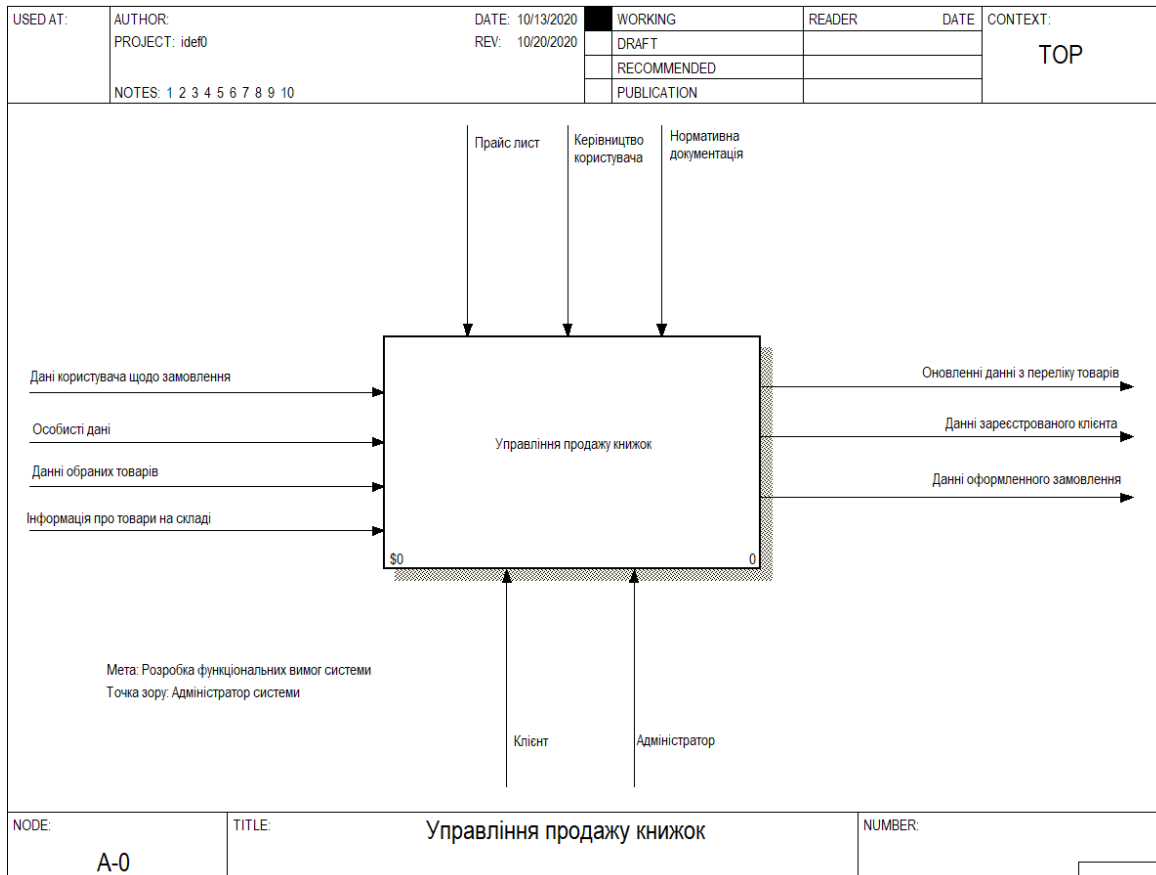


Рисунок 3.1 – Концептуальна діаграма інформаційної системи

На концептуальній діаграмі ми бачимо, що основний блок – це «Управління продажу книжок». Головна мета: розробка функціональних вимог системи. Вхідні стрілки: «Дані користувача щодо замовлення», «Особисті дані», «Данні обраних товарів», «Інформація про товари на складі». Керуючі для управління продажами - це «Прайс лист», «Керівництво користувача» та «Нормативна документація». У ролі механізмів виступають адміністратор і клієнт. Таким чином ми описали та задали усі необхідні параметри системи. На виході ми отримуємо: «Оновлені дані з переліку товарів», «Дані зареєстрованого клієнта» та «Дані оформленого замовлення».

Але так ми описали загальну схему роботи нашої системи. Тому, щоб дізнатися з яких процесів складається наш головний блок, ми декомпозиємо його.

Основна функція, представлена на контекстній діаграмі верхнього рівня, може бути розкладена на основні підфункції допомогою створення дочірньої діаграми. У свою чергу, кожна з цих підфункцій може бути розкладена на складові частини за допомогою створення дочірньої діаграми наступного, більш низького рівня, на якій деякі або всі функції також можуть бути розкладені на складові частини. Кожна дочірня діаграма містить дочірні блоки і стрілки, що забезпечують додаткову деталізацію батьківського блоку. Дочірня діаграма, створювана при декомпозиції, охоплює ту ж область, що і батьківський блок, але описує її більш детально. На рис. 2.2 представлена декомпозиція концептуальної діаграми. Вона дозволяє уточнити основну функцію системи.

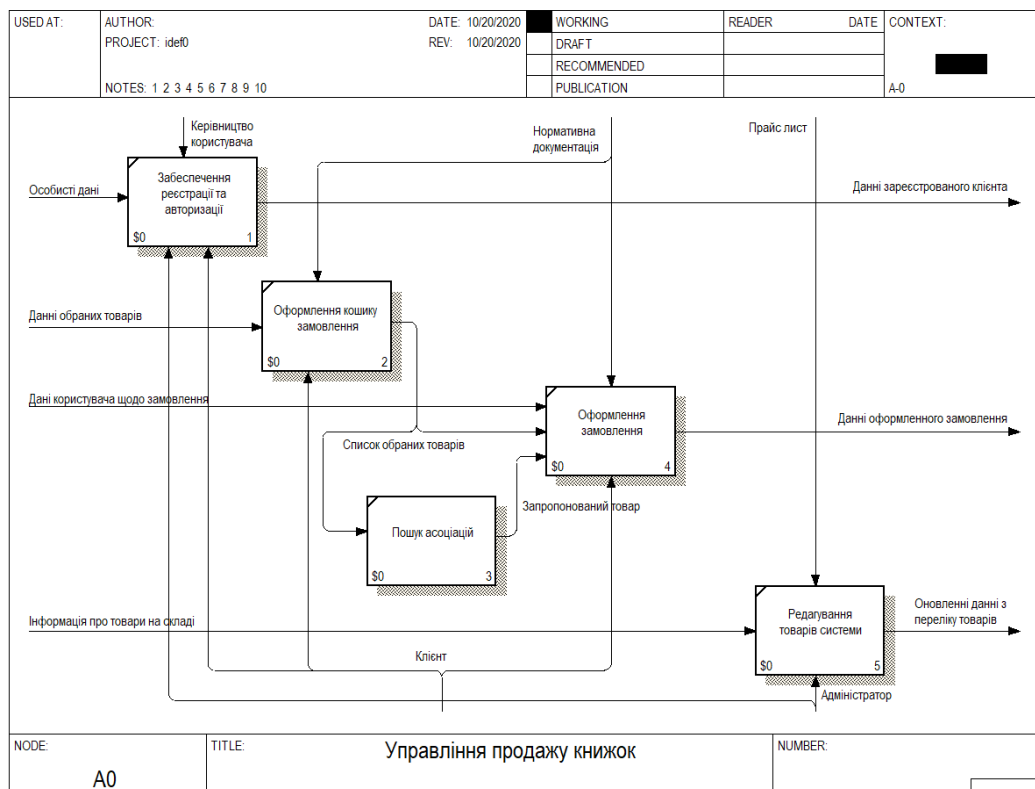


Рисунок 3.2 – Діаграма декомпозиції концептуальної діаграми

Як ми бачимо, у нашому випадку концептуальна діаграма складається з п'яти основних етапів

- забезпечення реєстрації та авторизації;
- оформлення кошику замовлення;
- пошук асоціацій;
- оформлення замовлення;
- редагування товарів системи.

У самому верху розташований блок з назвою забезпечення реєстрації та авторизації. Він необхідний тому що тільки авторизовані користувачі можуть оформляти замовлення. Наступний блок - оформлення кошику замовлення. Після підтвердження усіх доданих товарів, користувач переходить до оформлення замовлення. Але між цими блоками відбувається ще одна функція системи - пошук асоціацій. Він якраз і складається з розробленого метода веб майнінгу. Він необхідний для того, щоб знайти товар, який б можна була пропонувати клієнту. І останній функціональний блок - редагування товарів системи, необхідний адміністратору для управління системою.

Таким чином, проведене функціональне моделювання дозволило уточнити функціональні вимоги до розроблюваної системи.

3.2 Розробка моделі потоків даних

Наступним етапом, при визначенні вимог до системи, є створення діаграми потоків даних DFD, які використовуються для уявлення механізмів передачі та обробки інформації у системі, яка моделюється.

DFD діаграми дозволяють у простій та зрозумілій графічній формі відобразити як переміщується інформація у межах процесів системи. Головні елементи таких діаграм – це зовнішні сутності, процеси, сховища даних та потоки даних.

Зовнішні сутності - зовнішні системи, з яких надходить або куди прямує інформація в результаті взаємодії з зображеною системою. Іншими словами, це джерела і пункти доставки інформації, яка приходить або йде з системи. Такими сутностями можуть бути зовнішні організації, особи, комп'ютерні або бізнес-системи.

Процеси - будь-які процеси, які ведуть до зміни інформації та створення вихідних даних. Наприклад, виконання підрахунків, сортування даних відповідно до встановленої логіки або напрямом інформаційного потоку відповідно до бізнес-правилами.

Сховища даних - файли або репозиторії, де зберігається інформація для подальшого використання, наприклад, бази даних або форми заявки на участь.

Потоки даних - маршрути, за якими інформація переміщується між зовнішніми сутностями, процесами і сховищами даних. Потоки даних ілюструють взаємодію між іншими компонентами і відображаються у вигляді стрілок.

DFD діаграма верхнього рівня також називається контекстною схемою. Це найпростіший спосіб зображення аналізованих або модельованих систем і процесів. Такі схеми показують загальну картину і представляють систему у вигляді єдиного процесу, наділеного зв'язками із зовнішніми сутностями. такі діаграми так само можна декомпонувати. Декомпозиція дозволяє розбити узагальнений процес контекстної схеми на підпроцеси, тим самим можна побачити основні її функції. Процес декомпозиції зазвичай триває до третього або четвертого рівня, тому що далі декомпозиція хоч і дозволить побачити систему більш детально, але це її сильно ускладнить і буде складно розібратися в діаграмі і як то її аналізувати.

На рис. 3.3 представлена концептуальна діаграма DFD, яка відображає погляд на функції системи з боку її адміністратора.

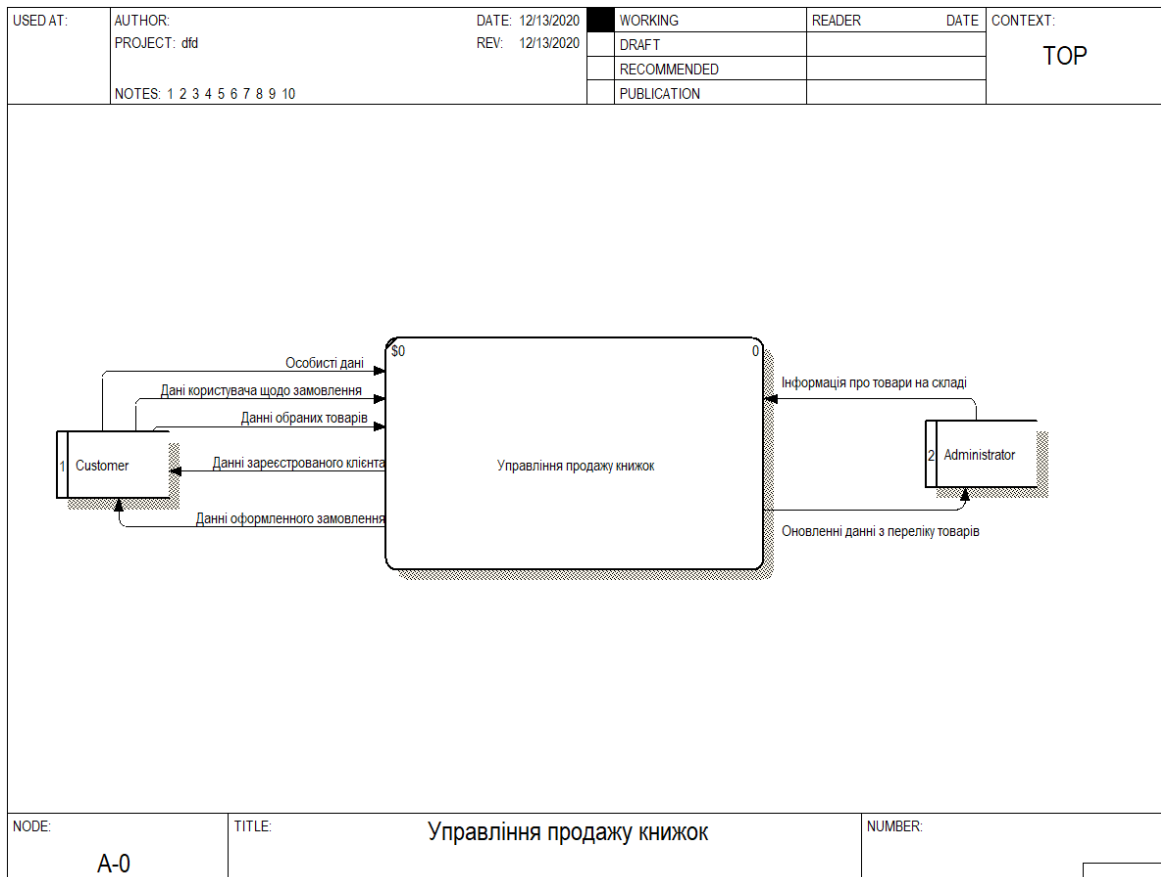


Рисунок 3.3 – Концептуальна діаграма потоків даних

Зовнішня сутність є матеріальним об'єктом поза контекстом системи, що є джерелом або приймачем системних даних. Передбачається, що об'єкти, що представлені такими вузлами, не повинні брати участь ні в якій обробці.

На концептуальній діаграмі ми бачимо, як переміщуються данні. Там розташовані дві сутності: адміністратор та клієнт. Від клієнта виходять стрілки, з особистими даними, дані користувача щодо замовлення та данні обраних товарів. Отримує він дані зареєстрованого клієнта та інформацію про оформлене замовлення. Адміністратор вводить інформацію про товари на складі і отримує оновлені данні з переліку товарів. Тепер ми декомпозуємо концептуальну діаграму потоків даних, щоб більш детально побачити як само переміщуються дані та між якими сутностями. На рис. 3.4 представлена декомпозиція концептуальної діаграми DFD.

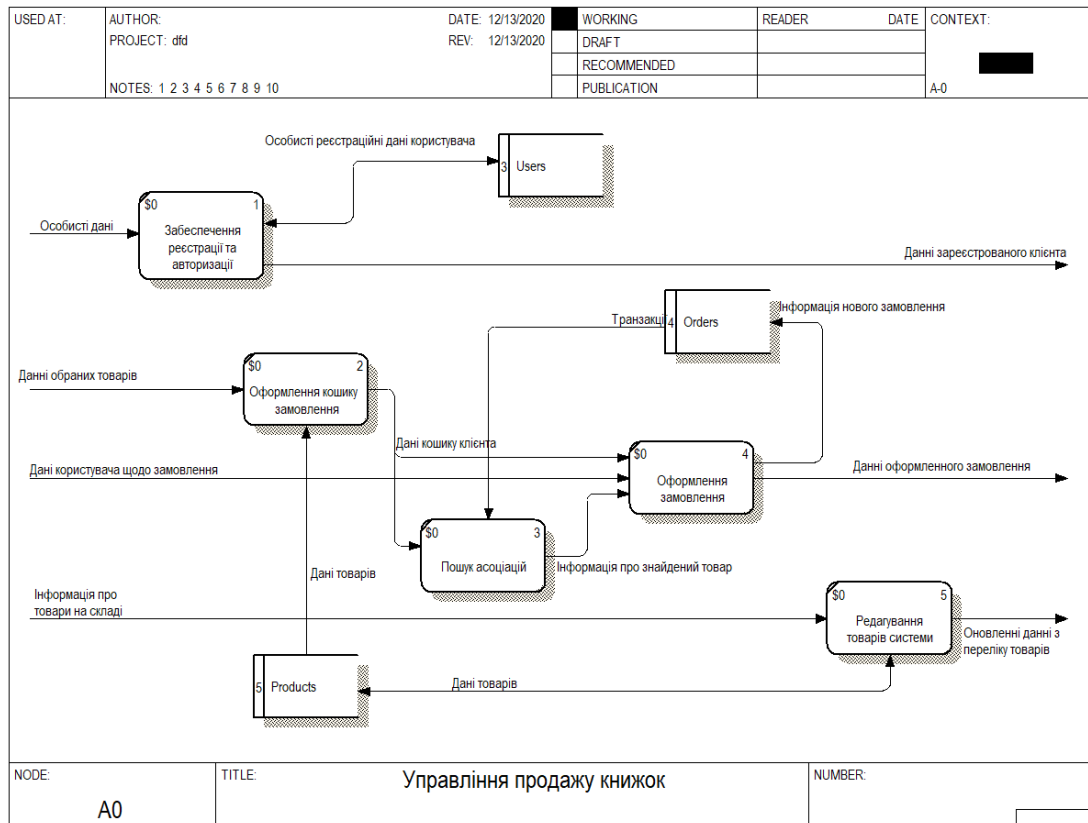


Рисунок 3.4 – Діаграма декомпозиції потоків даних

Опишемо як переміщуються інформація. Особисті дані потрапляють до блоку забезпечення реєстрації та авторизації. Цей функціональний блок зв'язується з сутністю Users, у якій зберігається інформація користувачів. Тому при реєстрації новий користувач додається а при авторизації перевіряється на наявність. На виході дані зареєстрованого клієнта. Наступний потік даних це обрані товари, які клієнт додав до кошика. Вони потрапляють до блоку оформлення кошику замовлення. Тут вони обробляються та переходять до блоків пошук асоціацій та оформлення замовлення. У першому використовується розроблений метод веб майнінга, результат якого повертається до другого блоку. До другого блоку також додається додаткова інформація користувача щодо замовлення. На виході ми одержуємо інформацію оформленого замовлення. Тут була задіяна сутність Orders, у якій зберігається інформація транзакцій усіх клієнтів. Вони необхідні для пошуку

асоціацій та додаються нові після оформлення замовлення. І останній потік даних пов'язаний з інформацією про товари на складі. Ці дані входять до блоку редагування товарів системи, а виходять вже оновлені дані переліку товарів. Сутність Products необхідна для зберігання інформації товарів у системі.

Таким чином, проведене моделювання потоків даних дозволило уточнити функціональні вимоги до розроблюваної системи, визначити склад таблиць, що входять в структуру БД.

3.3 Розробка діаграми варіантів використання

Для розробки діаграм використання скористаємося уніфікованою мовою моделювання UML - це мова для візуалізації, специфіцирования, конструювання та документування програмних систем.

UML - це мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення, моделювання бізнес-процесів, системного проектування та відображення організаційних структур.

Мова UML складається з трьох елементів: сутності - це абстракції, які є основними елементами моделей; відношення - це зв'язки між сутностями; діаграми - відображення взаємодії сутностей і відносин.

Сутності у свою чергу теж поділяються на чотири типи. Перші це структурні, які називають іменниками і зазвичай є компонентами або інтерфейсом користувачів системи. Другі це сутності для вираження поведінки. Їх правильно називати дієсловами і описують поведінку системи. Всього їх два типи, для обміну повідомленнями та послідовності станів. Наступні це анотаційні, які є пояснювальними для системи, наприклад примічення або коментарії до елементів моделі. І останні групуючі сутності - організуючі частини моделі, для організації елементів системи.

Існують ще чотири типи відносин в UML: залежності – це відношення між двома сутностями, коли зміни у одному призведуть до змін у другому;

узагальнення – коли спеціалізований об'єкт може бути представлений замість узагальненого; асоціація - співвідношення, яке описує семантичний зв'язок між об'єктами; композиція - різновид агрегації, де взаємозв'язок частини з цілим ще більш сильніша; реалізація - відношення між класифікаторами, при якому один визначає зобов'язання, а другий здійснює його виконання.

Основна ідея діаграм варіантів використання полягає в тому, що проєктована система представляється у вигляді безлічі акторів, що взаємодіють з системою за допомогою так званих варіантів використання. При цьому актором називається будь-який об'єкт, суб'єкт або система, що взаємодіє з моделюється системою ззовні. У свою чергу варіант використання - це специфікація функцій, які система надає актору.

На рис. 2.5 представлена діаграма варіантів використання, яка показує роботу системи з боку її різних типів ролей.

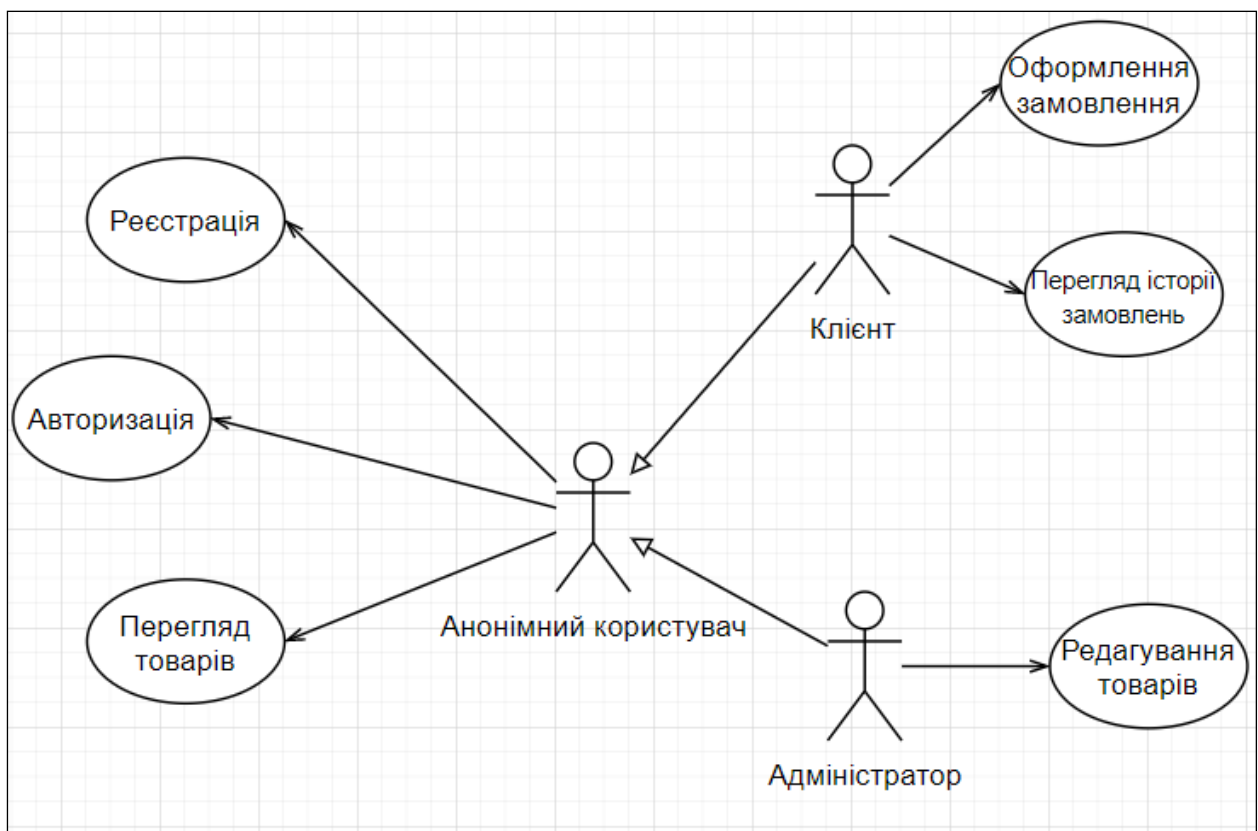


Рисунок 3.5 – Діаграма використання системи

На діаграмі зображено три актора – адміністратор, анонімний користувач та клієнт. Анонімний користувач може зареєструватися або авторизуватися та переглянути доступні товари. Клієнт – це зареєстрований користувач. Для нього відкривається декілько нових можливостей використання системи. Перша – це оформлення замовлення і друга – перегляд історії замовлень. Адміністратору також доступні усі можливості незареєстрованого користувача, плюс редагування товарів. Отже у результаті ми представили проєктовану систему у вигляді акторів, які взаємодіють з системою за допомогою прецедентів.

3.4 Розробка діаграма послідовності дій

Розробимо діаграму послідовності дій, щоб описати з якими компонентами інтерфейсу буде взаємодіяти наш користувач, які керують компонентами нам знадобляться, що ми будемо зберігати та якими повідомленнями будуть обмінюватися користувач і компоненти системи для виконання функції.

Основними елементами діаграми є взаємодіючі об'єкти з різними елементами і зв'язку між ними - взаємодіючі об'єкти обмінюються між собою деякою інформацією.

Об'єкти розташовані у горизонтальній послідовності, між ними передаються повідомлення. Вісь часу орієнтована зверху вниз. Елемент актор може використовуватися для подання користувача, що ініціює потік подій.

Кожен об'єкт має пунктирну лінію, час життя об'єкта, де цей елемент існує і потенційно бере участь у взаємодіях. Фокус управління позначається прямокутником на лінії життя об'єкта.

Повідомлення, якими обмінюються об'єкти, можуть бути декількох типів, повідомлення також можуть бути налаштовані для відображення операцій і властивостей вихідного та цільового елементів.

Елементи, такі як кордони, елементи управління і сутності, можуть використовуватися для моделювання користувальницького інтерфейсу, контролерів і елементів бази даних.

На рис. 3.6 представлена діаграма послідовності дій для «оформлення замовлень».

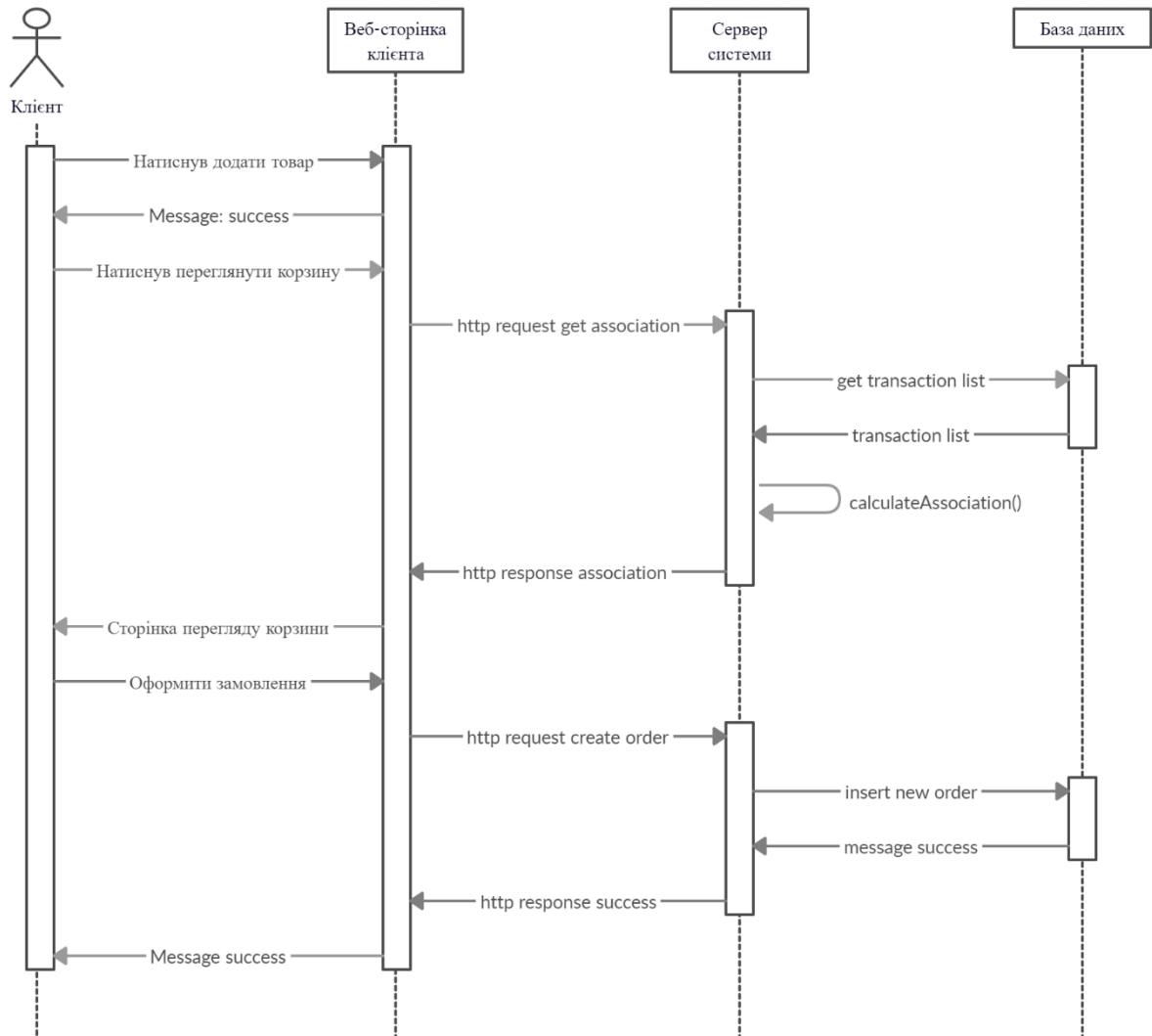


Рисунок 3.6 – Діаграма послідовності дій оформлення замовлення

На діаграмі зображена послідовність дій для оформлення замовлення клієнтом. Отже клієнт знаходиться на сайті та натискає на кнопку додати товар до кошику. Веб сторінка обробляє подію та відображає товар, що він потрапив

до кошику. Коли клієнт закінчив додавати товари, він натискає кнопку переглянути кошик. Веб сторінка переадресовує клієнта на сторінку перегляду кошика та посилає запит до серверу, щоб визначити який продукт пропонувати клієнту. Сервер посилає запит на отримання списку транзакцій та викликає розроблений метод. Після цього сервер посилає відповідь на запит, та веб браузер відображає список товарів у кошику та пропонування щодо перегляду ще одного товару. Далі клієнт може додати цей пропонований товар до кошику, заповнити усі необхідні форми та натиснути кнопку оформлення замовлення. Веб браузер знову посилає запит до серверу на створення нового замовлення. Сервер опрацьовує його та заносить то бази даних нову транзакцію та повертає браузеру відповідь, що операція пройшла успішно. Веб браузер відображає клієнту, що замовлення створено.

3.5 Розробка карти сайту

Карти сайтів допомагають визначитися зі структурою сайтів і додатків. Представлені на них ієрархії і зв'язку дозволяють клієнтам зрозуміти, де вони зможуть знайти потрібний матеріал.

Проста карта сайту зазвичай поміщається на одному аркуші паперу і схожа на звичайні організаційні діаграми. Карти будуються не тільки для веб сайтів; вони можуть використовуватися в будь-яких додатку пах, в яких доречно ідентифікувати сторінки, уявлення, стану і будь-яку інформацію, що виводиться. На рис. 3.7 зображена розроблена схема карти сайту.

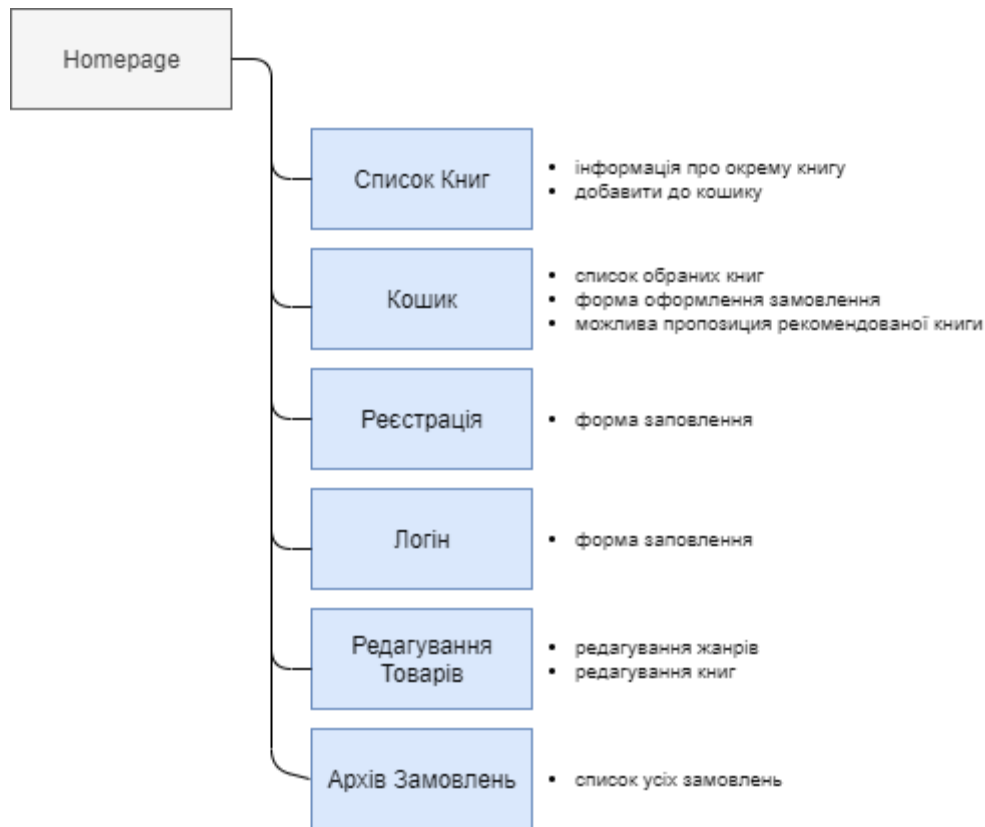


Рисунок 3.7 – Карта сайту

На рисунку зображено схему карти сайту. На ній присутні шість сторінок. Головна сторінка містить список книг, на якій є інформація про кожну окрему книгу та кнопки, для додавання їх до кошику. На сторінці кошику можна переглянути список товарів у кошику, форму для заповнення додаткової інформації щодо замовлення та повідомлення про пропонування ще одного товару. Сторінка реєстрації містить форми для заповнення особистих даних клієнта. На сторінці авторизації – форма для заповнення пошти та паролю. Сторінка редагування товарів надає адміністратору інструмент з редагування картинки та опису товарів. І остання сторінка – це архів замовлень, на якій користувач переглядає історію замовлень.

4 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ

4.1 Архітектура Model View Controller

Model View Controller переводиться як модель вигляд контролер - спосіб організації коду, коли код розбивають на декілька блоків і кожен блок відповідає за вирішення своїх завдань. Перший блок відповідає за дані системи, другий за відображення, а третій контролює роботу системи. Розглянемо роботу кожного блоку більш детально.

Блок Model відповідає за структуру системи, включає до себе дані і методи роботи з цими даними, реагує на запити з контролера, повертаючи дані або змінюючи свій стан. При цьому модель не містить в собі інформації про способи візуалізації даних або форматах їх подання, а також не взаємодіє з користувачем безпосередньо.

Блок View забезпечує відображення інформації. Ті ж самі дані можуть представлятися різними способами і в різних форматах. Наприклад, колекцію об'єктів можна уявити на рівні користувача інтерфейсу як в табличному вигляді, так і списком.

Controller забезпечує зв'язок між користувачем і системою, використовує модель і уявлення для реалізації необхідної реакції на дії користувача. Як правило, на рівні контролера здійснюється фільтрація отриманих даних і авторизація - перевіряються права користувача на виконання дій або отримання інформації.

Використання такого підходу спрощує розробку програмних засобів. Перевагою моделі MVC є чітке розділення проблем. Кожен блок шаблону MVC відповідає за чітко визначений аспект програми. У більшості програм немає плутанини щодо того, що належить до блоку вигляду або моделі.

У той час як контролери часто не можна використовувати багаторазово, об'єкти вигляду та моделі легко використовувати повторно. Якщо шаблон

MVC правильно реалізований, блок вигляду та моделі повинні складатися з компонентів, що використовуються багаторазово.

Головні переваги MVC:

Одночасна розробка - оскільки MVC роз'єднує різні компоненти програми, розробники можуть паралельно працювати над різними компонентами, не впливаючи та не блокуючи один одного.

Багаторазове використання - Один і той же вигляд для однієї програми може бути реконструйований для іншої програми з різними даними, оскільки подання просто обробляє те, як дані відображаються користувачеві.

Покращена масштабованість - якщо у програми виникають проблеми з продуктивністю, оскільки доступ до бази даних є повільним, ви можете оновити апаратне забезпечення, що запускає базу даних, без впливу інших компонентів.

Низький зв'язок між елементами - структура MVC така, що між моделями, видами або контролерами існує низька зв'язок.

Краща масштабованість - оскільки компоненти мають низьку залежність один від одного, внесення змін до одного (для виправлення помилок або зміни функціональності) не впливає на інші.

MVC - підхід до проектування додатки, який передбачає групування коду у блоки: модель, вигляд і контролер. Контролер обробляє вхідні запити. Модель дістає з бази даних інформацію, потрібну для виконання конкретних запитів. Вигляд визначає результат запиту, який отримує користувач.

4.2 Обґрунтування вибору мови програмування

Так як наша система використовувати патерн MVC, який розбиває систему на три блоки, то кожен блок може бути реалізований на різних мовах програмування. Блок модель – це реалізація баз даних у розроблюваній системі. Для цього будемо використовувати MongoDB - це програма

управління базами даних NoSQL з відкритим кодом. NoSQL використовується як альтернатива традиційним реляційним базам даних. Бази даних NoSQL досить корисні для роботи з великими наборами розподілених систем.

MongoDB використовує записи, які складаються з документів, що містять структуру даних, які складаються з пар полів та значень. Документи є основною одиницею даних у MongoDB. Поля в цих документах подібні до стовпців реляційної бази даних. Значення можуть містити різні типи даних, включаючи інші документи, масиви та масиви документів. Документи також включатимуть первинний ключ як унікальний ідентифікатор.

Набори документів називаються колекціями, які функціонують як еквівалент реляційних таблиць баз даних. Колекції можуть містити дані будь-якого типу, але обмеження полягає в тому, що дані в колекції не можуть бути розподілені по різних базах даних.

Як і інші бази даних NoSQL, MongoDB не вимагає заздалегідь визначених схем. Він зберігає будь-який тип даних. Це надає користувачам гнучкість для створення будь-якої кількості полів у документі, що полегшує масштабування баз даних MongoDB порівняно з реляційними базами даних.

Однією з переваг використання документів є те, що ці об'єкти відображаються у власних типах даних у ряді мов програмування. Крім того, вбудовані документи зменшують необхідність об'єднання баз даних, що може зменшити витрати.

Основною функцією MongoDB є горизонтальна масштабованість, що робить її корисною базою даних для компаній, що використовують додатки для великих даних. Крім того, шардінг дозволяє базі даних розподіляти дані по кластеру машин.

Вигляд буде реалізований на ReactJS. Одним з найважливіших аспектів React є той факт, що ви можете створювати багаторазові елементи HTML для швидкої та ефективної побудови користувальницьких інтерфейсів. React

простий у вивченні та у використанні, а також постачається документація, навчальні посібники та навчальні ресурси.

React базується на компонентах. Кожен компонент має свою логіку та керує власним рендерингом і може бути використаний повторно там, де вони вам потрібні. Повторне використання коду допомагає спростити розробку та підтримку додатків.

React спрощує створення динамічних веб-додатків, оскільки вимагає менше кодування та пропонує більше функціональних можливостей, на відміну від JavaScript, де кодування часто дуже швидко ускладнюється.

І останій блок це контроллер, який буде використовувати Node.js. Node.js - це міжплатформене середовище виконання з відкритим кодом для розробки серверних та мережевих додатків. Нижче наведено деякі важливі функції, які роблять Node.js першим вибором архітекторів програмного забезпечення.

Асинхронні - всі API бібліотеки Node.js є асинхронними, тобто не блокуючими. По суті, це означає, що сервер на основі Node.js ніколи не чекає, поки API поверне дані. Сервер переходить до наступного API після його виклику, і механізм сповіщення про події Node.js допомагає серверу отримати відповідь від попереднього виклику API.

Швидкість - побудована на механізмі JavaScript V8 від Google Chrome, бібліотека Node.js дуже швидко виконує код.

Однопоточна, але дуже масштабована - Node.js використовує однопоточну модель із циклічним подією. Механізм подій допомагає серверу реагувати неблокуючим способом і робить сервер дуже масштабованим на відміну від традиційних серверів, які створюють обмежені потоки для обробки запитів. Node.js використовує одну потокову програму, і ця сама програма може надавати послуги набагато більшій кількості запитів, ніж традиційні сервери, такі як Apache HTTP Server.

Без буферизації - програми Node.js ніколи не буферизують будь-які дані.
Ці програми просто виводять дані шматками.

4.3 Розробка моделі

При розробці діаграм потоків даних, було виділено три головні сутності: клієнт, продукт та замовлення. Але для реалізації такої моделі знадобиться ще одна допоміжна сутність – для зв'язку замовлень та продуктів. Розроблена модель приведена на рис. 4.1

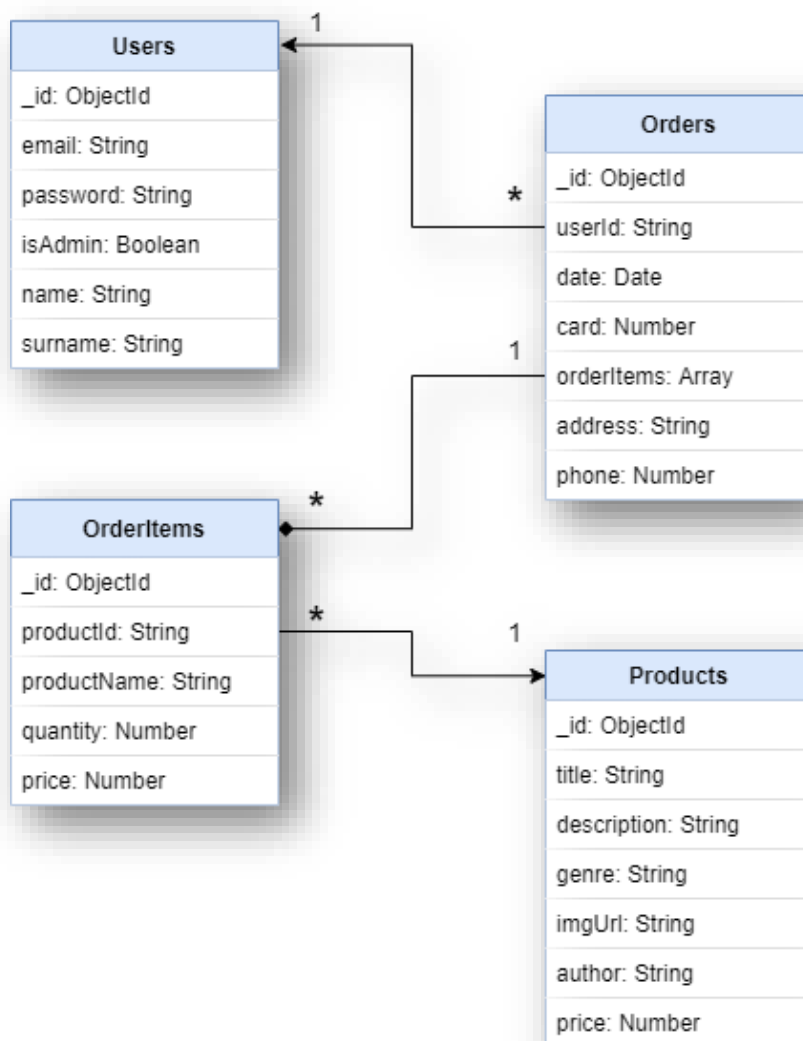


Рисунок 4.1 – Модель системи

«Users» зберігає дані користувачів, такі як логін пароль ім'я. У «Orders» зберігається інформація про замовлення, та містить такі поля, як дату замовлення, якому користувачу, адреса замовлення та контактний телефон. Також містить інформацію про те, які продукти були замовлені - «OrderItems», що описує товари у замовленні, їх ціну та кількість. І остання «Products» - містить інформацію про продукти у системі. Так як наша модель буде реалізована на MongoDB, то всі сутності будуть зберігатися у вигляді документів. Тому для прискорення роботи системи, ми об'єднаємо «Orders» та «OrderItems», тобто сутність з замовленнями буде зберігати не зовнішній ключ до продуктів у замовленні, а їх список. Завдяки цьому не треба буде робити два запити, а лише один. Детальніше опишемо кожен сутність у таблиці 4.1

Таблиця 4.1 – Сутності моделі системи

№	Найменування	Найменування атрибута	Тип даних (Домен)	Призначення
1	Users	email	текст	логін
		password	текст	пароль
		isAdmin	булева змінна	чи адміністратор
		name	текст	ім'я
		surname	текст	прізвище
2	Orders	userId	текст	чиє замовлення
		date	дата	дата замовлення
		card	не ціле число	карта оплати
		orderItems	ціле число	продукти замовлення
		address	текст	адреса
		phone	текст	телефон
3	Products	title	текст	назва
		description	текст	опис
		genre	текст	жанр
		imgUrl	текст	шлях до картинки

		author	ціле число	автор
		price	ціле число	ціна
4	OrderItems	productId	текст	ідентифікатор
		productName	текст	назва
		quantity	ціле число	кількість
		price	ціле число	ціна

За зазначеними описами моделі та типів даних, розробимо програмну реалізацію моделі. Продемонструємо на прикладі саму складну за зберіганням інформації – сутність Orders. Нижче приведено код розробленої сутності.

Лістинг 4.1 – Програмна реалізація сутності Orders

```
const orderSchema = new mongoose.Schema({
  userId: { type: String, required: true },
  date: { type: Date, default: Date.now() },
  card: { type: Number, required: true },
  orderItems: [
    {
      productId: { type: String, required: true },
      productName: { type: String, required: true },
      quantity: { type: Number, required: true },
      price: { type: Number, required: true },
    }
  ],
  address: { type: String, required: true },
  phone: { type: Number, required: true },
});
```

Останні дві сутності реалізовані за таким же принципом, але їх структура простіша. Усі необхідні функції для роботи з базами даних вже містяться у MongoDB, такі як знайти один або декілька записів, додати новий, або модифікувати існуючий. Залишилось лише викликати та правильно застосувати. Це вже робота контролера і буде описано у наступному розділі.

4.4 Розробка контролера

Контролер забезпечує зв'язок між користувачем і системою, використовує модель для реалізації необхідної реакції на дії користувача. Розіб'ємо контролер на окремі частини, кожен з яких буде обробляти свої запити. Працюватиме це таким чином, що для кожної моделі буде свій контролер. Тобто у нас буде три контролера – для товарів, користувачів та замовлень. Після реалізації контролерів, можна згенерувати діаграму класів, яка представлена нижче на рис. 4.2.

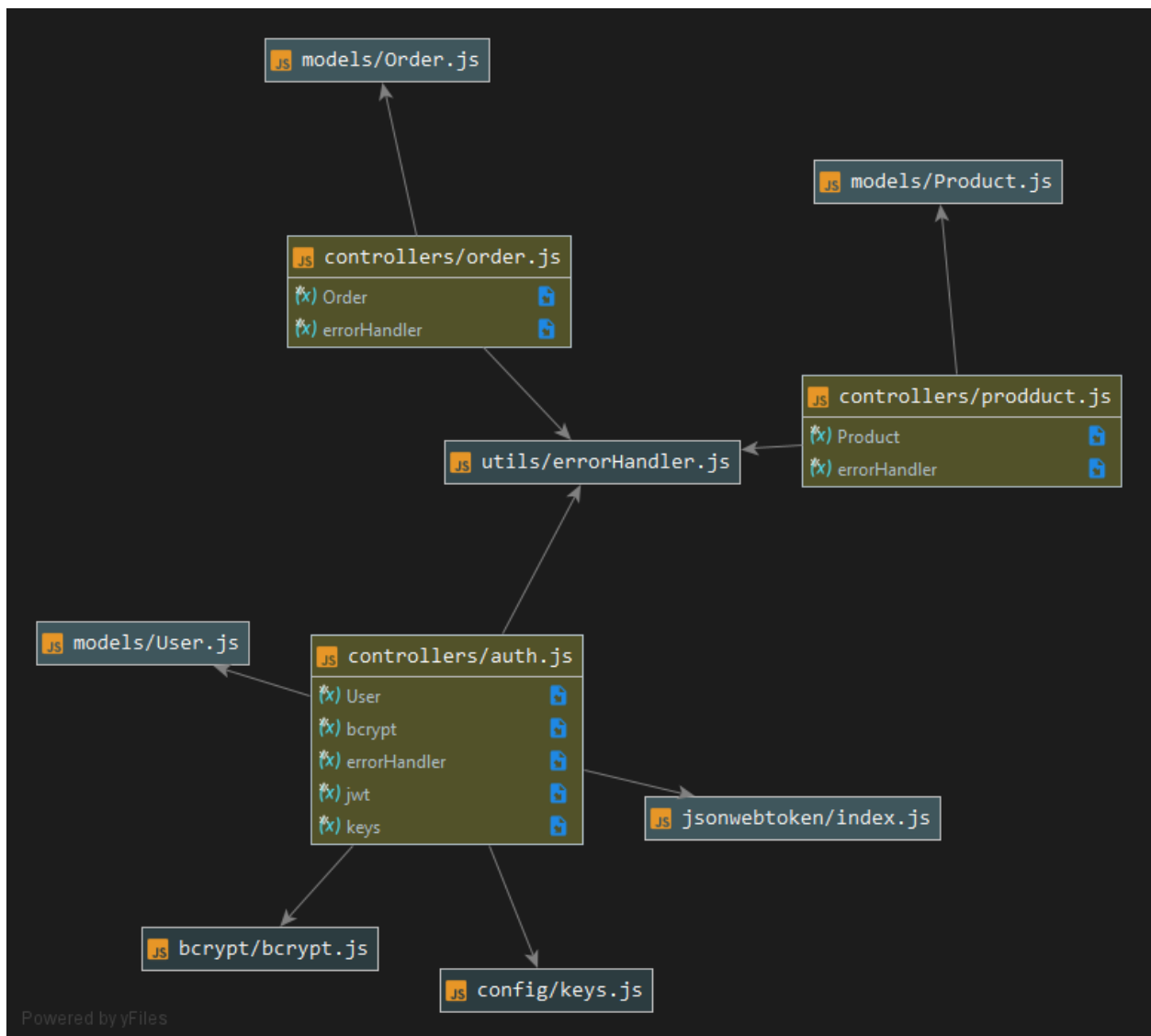


Рисунок 4.2 – Діаграма класів контролера

Опишемо які запити обробляються кожним контролером. Почнемо з того, що відповідаю за запити замовлень – `orders.js`. Для нього реалізовано три метода: знайти замовлення конкретного користувача, знайти замовлення за його `id` та створити нове замовлення. На прикладі продемонструємо роботу метода зі створення нового замовлення. Нижче приведено код методу.

Лістинг 4.2 – Програмна реалізація метода створення замовлення

```
module.exports.create = async (req, res) => {
  try {
    const order = await new Order({
      userId: req.user.userId,
      card: req.user.card,
      orderItems: req.body.orderItems,
      address: req.body.address,
      phone: req.body.phone,
    }).save();
    res.status(201).json(order);
  } catch (e) {
    errorHandler(res, e);
  }
};
```

Усі методи контролерів будуть реалізовані таким же чином. Контролер `auth.js` обробляє запити управління користувачами і реалізує методи створення нового користувача та логіну. І останній контролер `products.js` – відповідає за методи управління продуктами. Для нього реалізовано методи отримати перелік усіх продуктів, додати новий, видалити продукт за `id`, та відновити. За те, який метод викликати, відповідає роутер. Наприклад, якщо запит користувача був `../controllers/auth/login`, то він відправить обробляти таких запит контролеру `auth.js` та викликає метод входу до системи.

4.5 Розробка інтерфейсу

Інтерфейс реалізований за допомогою компонентів `ReactJS`. Дизайн сторінки дуже простий, але його легко модифікувати та масштабувати.

Виконаний інтерфейс у синіх тонах, має зверху панель управління, де можна переміщуватися між сторінками та кнопки входу виходу. Опишемо сторінку переліку товарів, всі інші сторінки матимуть схожий дизайн. По середині розташовані продукти, кожен продукт має своє зображення, назву. Нижче опис, зазначений автор та жанр. Ще нижче розташована ціна та кнопка додавання до кошика. На рис. 4.3 зображена головна сторінка з переліку товарів.

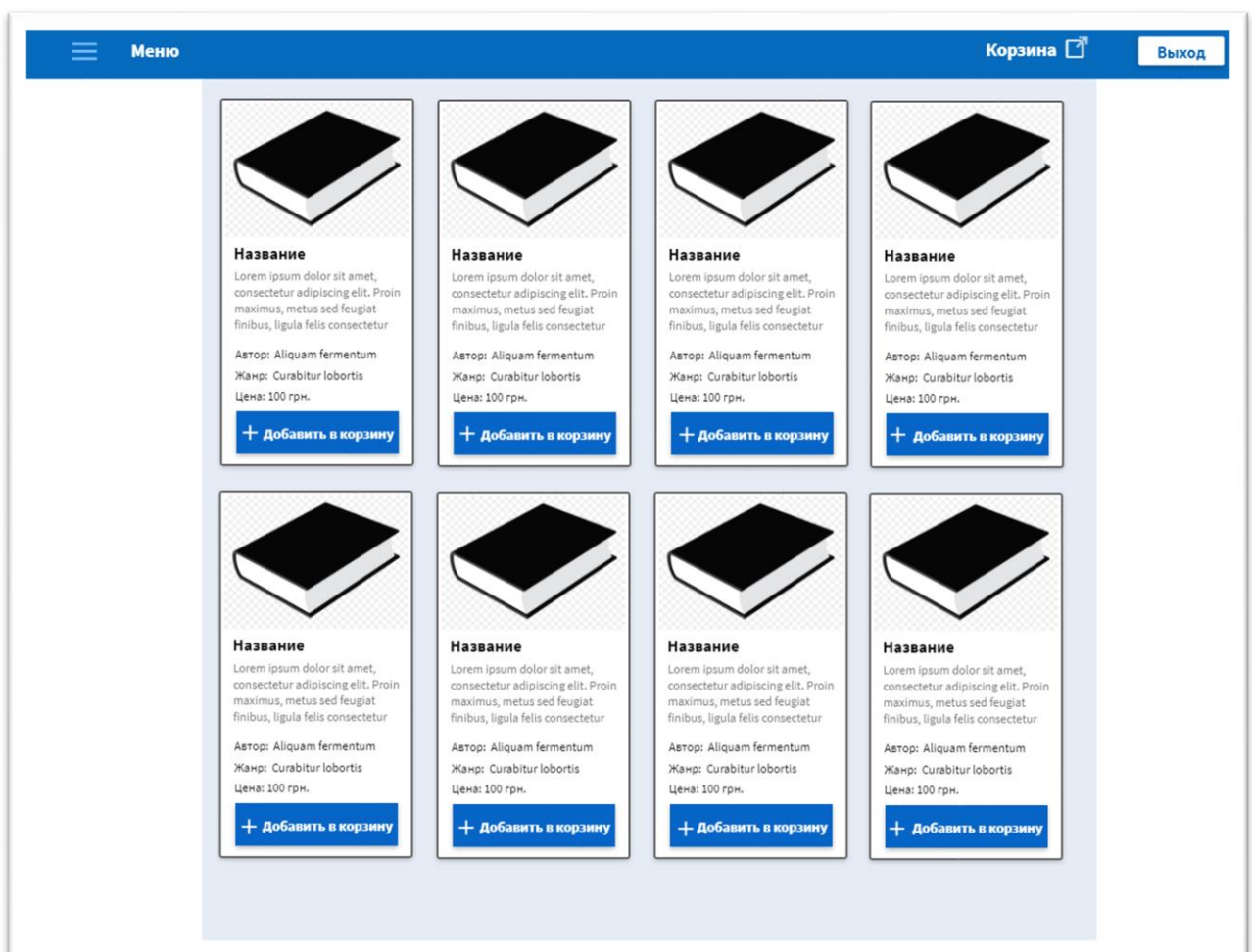


Рисунок 4.3 – Интерфейс головної сторінки

Отже інтерфейс реалізовано, модел та контролери теж. У купі отримуємо систему інтернет магазину з продажу книжок. Тепер можна перейти до впровадження методів до системи.

4.6 Впровадження розробленого метода

Як зазначалось раніше, розроблений метод веб майнінгу потрібен для пошуку товару, який можна було б пропонувати клієнту, коли він переглядає кошик та оформлює замовлення. На діаграмі послідовності дій розписано крок за кроком, як повинна реагувати система. Починається все с того що користувач відправляє запит на сервер, з інформацією про свої товари у кошику. Отже на сервері потрібно реалізувати контролер, який б обробляв цей запит. Для цього створемо окремий контролер, та назавемо його webMining у якого буде лише один метод – calculateAssociations, який і використовуватиме розроблений метод, та буде повертати результат його роботи. Браузер клієнта приймає відповідь та відображає товар, щоп пропонувати. Результат приведено на рис. 4.4.

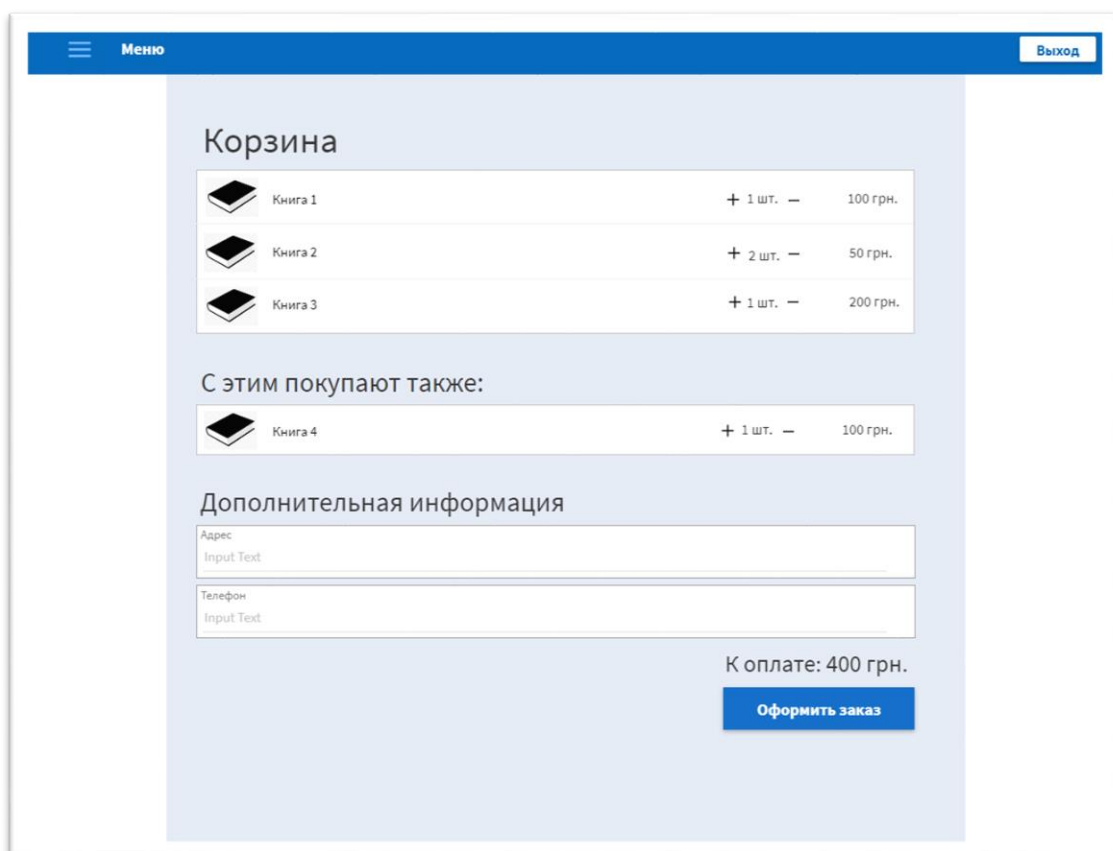


Рисунок 4.4 – Інтерфейс оформлення замовлення

ВИСНОВКИ

Під час виконання атестаційної роботи з розробки методу веб майнінга для видобутку знань про поведінку покупців книжкового магазину були розглянуті основні етапи розробки системи та метода.

На етапі аналізу розглянули що таке видобуток знань, описали які методи він використовує, було проведено аналіз електронної комерції та моделі поведінки покупців.

На етапі розробки методу веб майнінга розглянуто що таке веб майнінг, та був обраний метод для системи. Але обраний метод, метод пошуку асоціативних правил для аналізу ринкового кошику, мав недолік за головним критерієм системи – швидкість та простота роботи. Щоб отримати результат потрібно зробити багато розрахунків. Отже розроблений алгоритм використовує модернізований метод веб майнінга для пошуку асоціативних правил. У результаті алгоритм має гарні показники при використанні, які були проведені як на тестових даних, так і при впровадженні метода до системи.

Наступним кроком, щоб продемонструвати роботу метода, була розроблена система книжкового інтернет магазину. Була побудована діаграма IDEF0, щоб виділити функціональні вимоги, діаграма DFD, щоб дізнатися про те, як інформація переміщується у системі та які головні сутності сховища даних. Далі була розроблена діаграма варіантів використання, на який було виділено основні ролі користувачів системи та те, які функції вони можуть використовувати. Остання розроблена діаграма була потоків даних, для функції оформлення замовлення, яка продемонструвала як взаємодіє користувач із системою, як система крок за кроком обробляє запит, використовує розроблений метод та повертає результат. Завершення біло програмна реалізація системи та впровадження розробленого метода веб майнінга до неї.

Результат роботи – розроблена система книжкового магазину та модернізований метод веб майнінгу, який використовується для видобутку знань про поведінку покупців, рішення задачу аналізу ринкового кошику покупців.

ПЕРЕЛІК ПОСИЛАНЬ

1. Электронная коммерция: лабораторный практикум - Ю. В. Крутин; - Екатеринбург : РГППУ, 2018. - 103 с.
2. Тардаскіна Т.М. Електронна комерція: Навчальний посібник / Тардаскіна Т.М., Т19 Стрельчук Є.М., Терешко Ю.В. – Одеса: ОНАЗ ім. О.С. Попова, 2011. – 244 с
3. Ершов, К. С. Анализ и классификация алгоритмов кластеризации [Текст] / К. С. Ершов, Т. Н. Романова // Новые информационные технологии в автоматизированных системах. – 2016. – Вып. 19. – С. 274–279.
4. Шитиков В.К., Мастицкий С.Э. (2017) Классификация, регрессия и другие алгоритмы Data Mining с использованием R. 351 с.
5. Анализ данных и процессов: учеб. пособие / А. А. Барсегян, М. С. Куприянов, И. И. Холод, М. Д. Тесс, С. И. Елизаров. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2009. — 512 с.
6. Чернышева, Г. Ю. Интеллектуальный анализ данных [Текст] : учеб. Пособие для студентов специальности 080801.65 «Прикладная информатика (в экономике)» / Г. Ю. Чернышева. – Саратов : Саратовский государственный социальноэкономический университет, 2012. – 92 с.
7. Интеллектуальный анализ данных. Алгоритмы Data Mining : учебное пособие / Е. М. Герасименко; Южный федеральный университет. – Ростов-на-Дону – Таганрог : Издательство Южного федерального университета, 2017. – 84 с.
8. Поведение потребителей [Электроний ресурс] – Режим доступа: [www / URL: \[https://studme.org/48429/marketing/povedenie_potrebiteley\]\(https://studme.org/48429/marketing/povedenie_potrebiteley\)](http://www.studme.org/48429/marketing/povedenie_potrebiteley) - 12.05.2019 г. - Загол. з екрану.
9. Анализ рыночной корзины и ассоциативные правила [Электроний ресурс] – Режим доступа: [www / URL: <https://habr.com/ru/post/66016/>](http://www.habr.com/ru/post/66016/) - 11.02.2019 г. - Загол. з екрану.

10. О. Гуров. Розробка метода веб майнінга для видобутку знань про поведінку покупців книжкового магазину. Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем (MEICS-2020). Тези доповідей на V Всеукраїнській науково-практичній конференції: 25-27 листопада 2020 р., м. Дніпро / Укладачі Іванченко О. В., Вашерук О. В. – Дніпро, Дніпровський національний університет імені Олеся Гончара, Кременчук: ПП Щербатих О. В., 2020. – 218 с.

11. Sitnikov, D. Informativity of Association Rules from the Viewpoint of Information Theory / Sitnikov, D., Titova, O., Minukhin, S., Kovalenko, A., Titov, S. // International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, Proceedings. – 2019. – P. 595–598.

12. CASE-технологии: учеб. пособие / Д. В. Александров, И. В. Грачев, Д. Н. Фадин; Владим. гос. ун-т. – Владимир: Изд-во Владим. гос. ун-та, 2006. – 64 с.

13. Капон Н., Колчанов В., Макхалберт Дж. Управление маркетингом. [Электронный ресурс]. — Режим доступа: <https://getabook.pp.ua/books/uvpravlennie-marketingom>

14. Котлер Ф., Армстронг Г., Вонг В., Сондерс Дж. Основы маркетинга. [Электронный ресурс]. — Режим доступа: http://artlib.osu.ru/web/books/content_all/1917.pdf

15. Котлер Ф., Келлер К.Л. Маркетинг менеджмент. Экспресс-курс. — [Электронный ресурс]. — Режим доступа: https://nataliaakulova.ru/wpcontent/uploads/2015/02/01/Filipp_Kotler_Kevin_Keller_Marketing_menedzhmen.pdf

16. Катаев А.В. Маркетинг: сущность, определения и виды. [Электронный ресурс]. — Режим доступа: <http://kataev.ru/28/>

17. Sitnikov D. Assessment of extended aggregated association rules / Sitnikov D., Ryabov O., Titova O., Kovalenko A. // Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies,

DESSERT'2018, 24-27 May, 2018, Kyiv, Ukraine, page 95-99. (IEEE Xplore, Scopus)

18. Jirachai Buddhakulsomsiri, Warut Pannakkong, Suebsak Nanthavanij, "Application of association rule algorithm to industrial safety data mining," *International Journal of Industrial and Systems Engineering*, vol. 21, issue 4, 2015.

19. Amir A.; Feldman R.; Kashi R., "A new and versatile method for association generation," *Information Systems*, vol. 22, № 6/7, pp. 333-347, 1997.

20. D. Sitnikov, E Titova, O. Ryabov, "A method for association rule quality evaluation based on information theory," *WIT Transactions on Information and Communication Technologies*, pp. 25-34, 2006.

21. R.Srikant, R.Agrawal, "Mining generalized association rules," *Proc. of the 21th VLDB Conference Zurich, Switzerland*, pp. 407-419, September 1995.

22. Ситников Д.Э., Титова Е.В. Метод поиска обобщенных ассоциативных зависимостей между дискретными признаками // *Системы обробки інформації*. – Харьков: ПАНМ, ХВУ. – 2002. – Вып. № 6(22). – С.

23. Титова Е. В. Сравнительная характеристика простых и расширенных ассоциативных правил для признаков объектов в базах данных // *Системы обробки інформації* – Харьков: НАНУ, ПАНМ, ХВУ. – 2003. – № . – С. 31-37.

24. Ситников Д. Э., Титова Е. В. Влияние стандартных параметров ассоциативного правила на его информативность // *Системы обробки інформації* – Харьков: НАНУ, ПАНМ, ХВУ. – 2004. – № 4. – С.

25. Ситников Д.Э., Титова Е.В., Романенко О.А. Система поиска обобщенных ассоциативных правил в базах данных // *Збірник наукових праць Інституту проблем моделювання в енергетиці НАН України*. – К.: НАН України. – 2004. – Вып. 25. – С.