

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти – другий (магістерський)

Дослідження методів підтримки прийняття рішень при виборі товарів у
Інтернет-магазині при покупці
(тема)

Виконав: студент 2 курсу, групи ІПЗм-18-3
Бездітний А.Е.
(прізвище, ініціали)

спеціальності 121 - Інженерія програмного
забезпечення
(код і повна назва спеціальності)

Освітньо-наукової програми Інженерія
програмного забезпечення
(повна назва освітньої програми)

Керівник проф. Дудар З.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф.

(підпис)

Дудар З.В.
(прізвище, ініціали)

2020 р.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти - другий (магістерський)

Спеціальність 121 - Інженерія програмного забезпечення
(код і повна назва)

Тип програми освітньо-наукова програма

Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 20 ____ р.

ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

студентові Бездітний Артем Едуардович
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів підтримки прийняття рішень при виборі товарів у Інтернет-магазині при покупці.
затверджена наказом університету від «27» березня 2020 р. № 473 Ст
2. Термін подання студентом роботи до екзаменаційної комісії _____ 2020 р.
3. Вихідні дані до роботи електронні ресурси за обраною тематикою, мінімальні вимоги до функціональності програми, загальні вимоги до архітектури системи
4. Перелік питань, що потрібно опрацювати в роботі: аналіз предметної області, концептуальне моделювання предметної області, аналіз існуючих методів підтримки прийняття рішень, пошук методів мінімізації розмірності даних, розробка моделі задачі, розробка бази даних, реалізація СППР.
5. Консультанти розділів роботи

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка *
1	Аналіз проблемної області та постановка задачі	30 березня 2020 р	
2	Аналіз існуючих методів СППР	4 квітня 2020 р	
3	Аналіз та моделювання предметної області	8 квітня 2020 р	
4	Розробка математичної моделі предметної галузі	10 квітня 2020 р.	
5	Розробка схеми бази даних	11 квітня 2020 р	
6	Розробка алгоритму ППР	18 квітня 2020 р.	
7	Створення коду програми	2 травня 2020 р.	
8	Тестування і налагодження програми	5 травня 2020 р.	
9	Підготовка пояснювальної записки	12 травня 2020 р.	
10	Підготовка презентації та доповіді	16 травня 2020 р.	
11	Попередній захист	17 травня 2020.	
12	Нормоконтроль, рецензування	19 травня 2020 р.	
13	Занесення диплома в електронний архів	20 травня 2020 р.	
14	Допуск до захисту у зав. кафедри	20 травня 2020 р.	
* заповнюється вручну після виконання чергового пункту			

Дата видачі завдання _____ 2020 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Дудар З.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Атестаційна робота магістра містить: 85 с., 36 рис., 18 джер.

ІНТЕРНЕТ-МАГАЗИН, ПОКУПКИ, ПРИЙНЯТТЯ РІШЕНЬ, ТОВАР, РЕКОМЕНДАЦІЯ, ПРЕЦЕДЕНТИ, АНАЛІЗ ДАНИХ, JAVA.

Об'єктом дослідження є методи підтримки прийняття рішень та засоби реалізації програмних систем спрямованих на оптимізацію процесу здійснення покупок в інтернет-магазинах.

Метою роботи є розробка автоматизованої системи-помічника інтернет-магазину.

Методи розробки базуються на інструментах розробки програмних застосувань на платформі Java, фреймворку Spring і бази даних MongoDB.

В результаті роботи здійснена програмна реалізація системи-помічника в інтернет-магазині, яка представляє собою WEB-додаток на основі фреймворку SpringMVC.

WEB STORE, PURCHASES, DECISION, GOODS, RECOMMENDATION, CLUSTERIZATION, PRECEDENTS, DATA MINING, JAVA.

The object of the study is the methods and algorithms for constructing an optimal schedule of sports events.

The purpose of the work is to compile an algorithm for constructing the optimal schedule of sports competitions as well as the implementation of a system for recording sports achievements.

Development methods are based on tools for developing software applications on the Java platform, Spring framework and MongoDB database.

As a result of the work the algorithm of the construction of the optimal schedule of sports competitions and the accounting system, which is a WEB-application based on the framework of Spring, has been developed.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	8
1.1 Аналіз предметної області.....	8
1.2 Виявлення проблем та актуалізація рішень.....	11
1.3 Постановка задачі.....	15
2 ПЕРЕЛІК ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ.....	18
2.1 Призначення розробки.....	18
2.2 Вимоги до програмного продукту.....	19
3 ОПИС МЕТОДІВ, МОДЕЛЕЙ ТА АЛГОРИТМІВ.....	22
3.1 Аналіз та UML-моделювання предметної області.....	22
3.2 Аналіз існуючих методів підтримки прийняття рішень.....	27
3.3 Математичне моделювання.....	31
3.4 Розробка схеми бази даних для зберігання математичної моделі.....	33
3.5 Розробка алгоритму планування.....	36
3.6 Розробка архітектури застосунку.....	42
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ.....	45
4.1 Опис фізичної моделі бази даних.....	45
4.3 Реалізація MapReduce ноди.....	48
4.4 Інтерфейс системи.....	49
4.4.1 POST /data/event.....	49
4.4.2 POST /data/product-property.....	50
4.4.3 POST /data/category.....	50
4.4.4 POST /analyze.....	51

4.4.5 POST /recommend	51
5 АНАЛІЗ ДОСЛІДНИЦЬКОЇ ЕКСПЛУАТАЦІЇ ТА МОЖЛИВИХ ЗАСТОСУВАНЬ	53
5.1 Аналіз можливих застосувань	53
5.2 Опис тестування системи.....	54
ВИСНОВКИ.....	59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	60
ДОДАТОК А Слайди презентації.....	62
ДОДАТОК В Лістинг програмного коду.....	79

ВСТУП

Історія торгівлі, як обмін товарно-матеріальними цінностями, відома з кам'яного віку. Вже тоді вона існувала в звичному для нас значенні: пропозиція до обміну з метою отримання вигоди.

Спочатку торгівля була виключно натуральної та сталася вона, за однією з версій, з звичаю обмінюватися подарунками. Пізніше люди стали обмінюватися предметами рівноцінними, наприклад, молоток замість сокири або м'ясо тварини замість овочів або фруктів. Головними передумовами для подальшого розвитку торгівлі стала спеціалізація промисловості і монета, роль якої у різних народів грали коштовності, раби, хутра, худобу і т. д.

Протягом століть торгівля піддавалась безперервним вдосконаленням, з'являлись нові методи укладення торгових угод. Суттєвої подією в даній галузі стала поява персональних комп'ютерів, поява глобальної мережі Інтернет та, як наслідок, інтернет-магазинів. Це дозволило зробити процес покупки максимально комфортним. Споживачеві не доводиться витрачати час на пошук необхідних речей в фізичних торгових точках - він може відкрити сайт інтернет-магазину, де на віртуальних вітринах представлені будь-які товари, забезпечені докладним описом і характеристиками. Вибрати і замовити їх можна парою клацань комп'ютерної миші.

Зі збільшенням торгових оборотів, все більшого значення набувало ведення діяльності, спрямованої на досягнення цілей підприємств, установ, організацій шляхом формування попиту та максимального задоволення потреб споживачів. Наразі поява майже кожного нового продукту на ринку супроводжується проведенням масштабних маркетингових кампаній, що мають на меті переконати потенційних покупців в необхідності придбання певного продукту та його переваги над конкурентами[1].

Кількість найменувань, доступних в одному інтернет-магазині може перевищувати тисячі одиниць в різноманітних категоріях, від харчових продуктів

до транспортних засобів. Кожен товар має власний набір характеристик і зрідка певний вибраний продукт має абсолютну перевагу над конкурентним. Тож вибір найбільш релевантного продукту може бути непростю задачею, особливо, якщо взяти до уваги те, що не кожен покупець є компетентним спеціалістом і має мінімально необхідні знання для вибору необхідного товару.

Величезний асортимент товарів та широкий перелік їх характеристик зумовлює необхідність існування окремих спеціалістів, що мають допомогти у виборі товару, котрий найліпше відповідає потребам покупців. Перевагою та одночасно недоліком інтернет-магазинів є максимальна автоматизація процесів здійснення покупки, як наслідок – відсутність продавці-консультантів та проблеми в виборі бажаного товару.

Саме в таких випадках було б доцільним створення системи, покликаної замінити фізичного консультанта[2]. Подібна система може рекомендувати покупцям певні позиції на основі вже придбаних самостійно або іншими покупцями товарів, допомогти у виборі конкретного продукту серед представлених в асортименті аналогів. Існування такого автоматизованого помічника є взаємовигідним для покупця та продавця, для останнього вона дає можливість збільшити продажі шляхом пропозиції найліпших товарів для кожного окремо взятого покупця відповідно до його потреб.

Отже, метою атестаційної роботи є підготовка математичної моделі для обраної предметної галузі, на базі якої необхідно розробити алгоритм підтримки прийняття рішень з урахуванням існуючих методів та власних модифікацій. Розроблений алгоритм має бути використаний для програмної реалізації СППР та протестований на наборі даних реальних Інтернет-магазинів.

1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

Протягом останніх років електронна комерція швидко розвивається у всьому світі, забезпечуючи при цьому безпрецедентні засоби і можливості для розвитку світової торгівлі. Перші системи електронної комерції з'явилися понад 50 років тому у США для замовлення транспортних квитків. А вже у 90-х виникли перші інтернет-магазини, де люди могли купувати товари та оплачувати їх банківськими картками. Поява інтернету пришвидшила розвиток електронної комерції, адже вартість обміну інформацією стала значно меншою.

У багатьох випадках електронна комерція дозволяє скоротити шлях перепродажу продукту від виробника до споживача. Це можливо завдяки використанню Інтернет-технологій, що надають можливість ефективної прямої взаємодії з кінцевим споживачем, тому компанії можуть виконувати роль, яку традиційно виконували проміжні постачальники. Це також дозволяє накопичувати інформацію про усі продажі та про усіх клієнтів, що, у свою чергу, дозволяє виконати досконалий бізнес-аналіз та маркетингові дослідження. Це є великою перевагою у конкурентній боротьбі.

Обіг на українському ринку оцінюється орієнтовно у два мільярди доларів. Найчастіше українці купують онлайн одяг, аксесуари, подарунки, взуття, косметику, дрібну побутову техніку та смартфони. Також користуються популярністю автотовари та інвентар для дому та саду.

Орієнтовно 12-13 разів на рік українці щось купують в інтернеті. Середній чек складає 1200 грн, згідно з вибіркою кореспондентів компанії GFK за даними на 2018 рік. Тобто, середньостатистичний українець витрачає на онлайн-покупки 60 доларів на рік.

За інформацією компанії eMarketer, що спеціалізується маркетингових дослідженнях наразі спостерігається стабільний приріст обігу коштів в галузі електронної комерції. Так, на рис. 1.1 представлена інформація, що характеризує

позитивні тенденції щодо продажів через Інтернет у світі в період з 2015 по 2020 роки. Навіть якщо взяти до уваги те, що темпи росту поступово зменшуються, загальна частка в роздрібних продажах невинно зростає.



Рисунок 1.1 – Інфографіка обігу коштів та частки продажів для електронної комерції

Безперечно, Інтернет-магазини мають деякі спільні риси з класичними магазинами з фізичними точками продажу, та все ж вони відрізняються один від одного. Власник кожного магазину прагне зробити свій сайт максимально зручним для відвідувача, удосконалюючи систему замовлення та способи переходу від одного розділу до іншого. Як і в звичайному магазині, в Інтернет-магазині можуть влаштовуватися розпродажу і знижки. Головна відмінність Інтернет-магазину від звичайного магазину - це не тільки можливість купити щось, не виходячи з дому

чи офісу, а також можливість витратити менші кошти. За рахунок чого виходить так, що покупка в Інтернет-магазині стає краще.

На сьогодні власники Інтернет-магазинів нерідко прибігають до використання спеціальних збору статистики, щоб отримати чітку уяву про дії покупців на їх сайті та прийняти відповідні рішення для поліпшення свого сервісу. Однією з найбільш популярних подібних систем є Google Analytics — це зручний і багатофункціональний сервіс від компанії Google для аналізу інтернет-сайтів та мобільних додатків. Дозволяє веб-майстрам перевірити стан індексування, створити детальну статистику аудиторії сайту та оптимізувати видимість своїх веб-сторінок. В даний час платформа належить торговій марці Google Marketing Platform. Google Analytics є найбільш широко використовуваним сервісом веб-аналітики в Інтернеті.

Google Analytics використовується для відстеження активності користувачів на веб-сайті, такої, як тривалість сеансу, сторінки за сеанс, показник відмов тощо, а також інформації про джерело трафіку (рис. 1.2). Доступна інтеграція з Google Ads, яка дозволяє веб-майстрам, відстежувати якість цільової сторінки та конверсію. Аналіз Google Analytics дозволяє виявити неефективні сторінки, враховуючи кількість сеансів, тривалість перебування користувача на веб-сторінці, вірогідність конверсії. У сервісі також доступна функція власної сегментації відвідувачів.

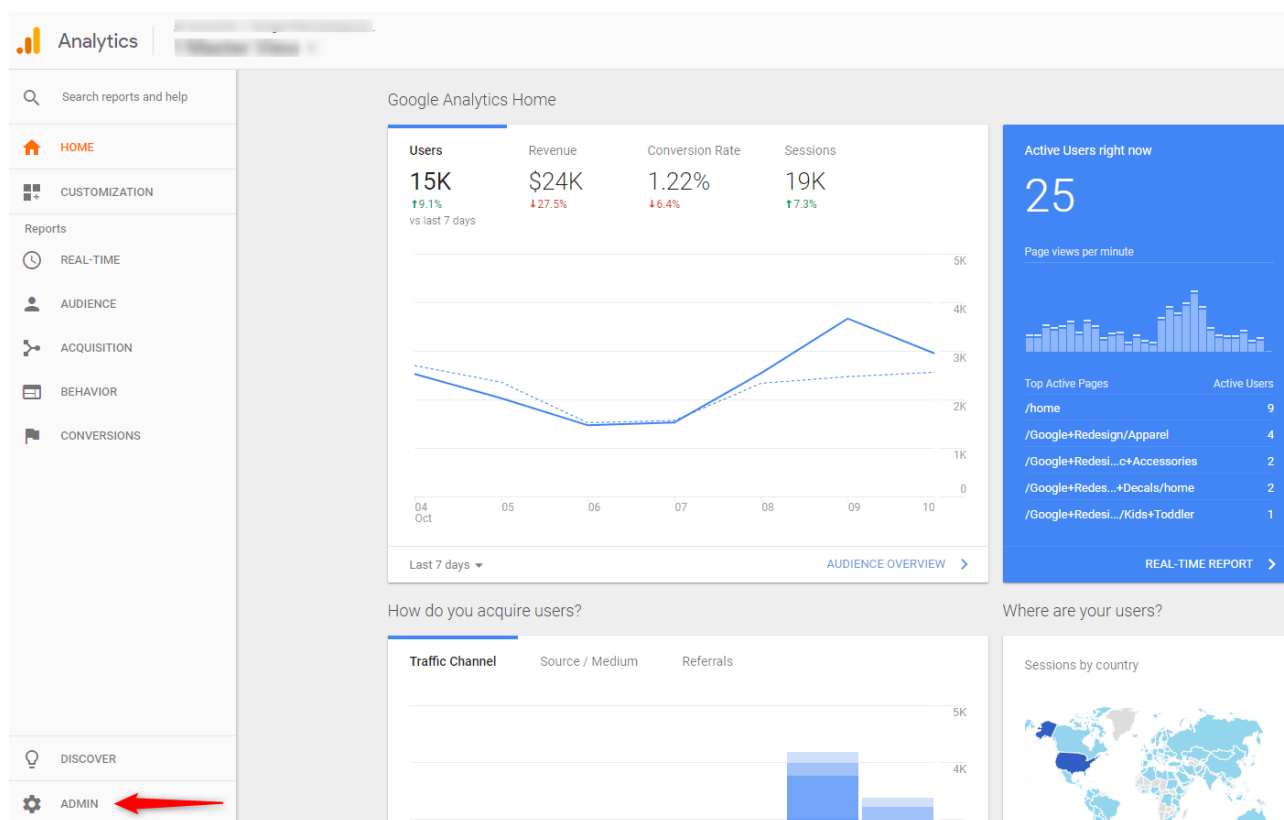


Рисунок 1.2 – Зібрана аналітика одного з Інтернет-магазинів сервісом Google Analytics

Мали місце декілька онлайн-дискусій про вплив Google Analytics на ефективність сайту. Однак Google представив асинхронний код JavaScript у грудні 2009 року, щоб зменшити ризик уповільнення завантаження сторінок, позначених сценарієм `ga.js`.

1.2 Виявлення проблем та актуалізація рішень

Незважаючи на те, що електронна комерція покликана полегшити здійснення покупок і постійні нововведення в цій галузі зумовлюють її стрімкий розвиток, вона не позбавлена недоліків. Попри всі переваги такого способу придбання товарів, все ж таки більша кількість торгових транзакцій здійснюється очно в

фізичних магазинах, на торговельних майданчиках. Частково ця різниця пояснюється новизною інтернет-магазинів, для людей старшого віку вони є диковинками, довіра до такого способу покупок недостатня, а для деяких категорій товарів варіант їх придбання через Інтернет може обернутися вибором невідповідного товару, наприклад, при пошуку одягу. Ці недоліки відносяться до таких, що не можуть бути усунені в найближчий час.

Окремо доцільно звернути уваги на недоліки інтернет-магазинів, що можуть бути усунені повністю або частково, як результат вдосконалення та автоматизації торгових процесів.

Асортимент інтернет-магазинів нерідко вражає уяву. З одного боку, це можливість для покупців обрати товар, який буде максимально задовольняти їх потреби, з іншого – зайва перепона, що зумовлює трату часу на пошук необхідного товару. Якщо необхідний час для вибору перевищує очікування потенційного покупця, він залишає дану торгову платформу. Тож немаловажним показником та однією з проблем є трата зайвого часу для вибору товару.

Кожен покупець для здійснення будь-якої покупки має володіти мінімальним набором знань про шукані продукти, а враховуючи неспинний розвиток технологій та ускладнення їх виробництва, об'єм необхідних знань лиш збільшується з плином часу. Певні інтернет-магазини вирішують цю проблему шляхом введення позиції онлайн консультанта, котрий має допомогти покупцю в виборі бажаного товару. Одним із найбільш популярних продуктів, що надають можливість подібного онлайн-консультування є розробка компанії «tawk.to inc» (рис. 1.3).

Офлайн ×

Будь ласка, заповніть нижче подану форму і ми вам дамо відповідь, як тільки це буде можливо.

* Ім'я

Телефон

* Email

* Повідомлення

Надіслати

We're ⚡ by tawk.to

Рисунок 1.3 – Форма швидкого зв'язку з онлайн-консультантом

Проте і подібне рішення має свої недоліки, зокрема, як правило, обмеженість в часі, тобто відсутність цілодобової підтримки, а також необхідність найму відповідних фахівців до штату, що тягне за собою додаткові фінансові трати. Втім, незаперечною перевагою такого рішення є можливість рекомендацій для покупця ґрунтуючись на його початкових побажаннях. Будь-яка програмна система в даному випадку передбачає існування певних даних на основі яких буде відбуватися прийняття рішень, тобто можливість рекомендації «нульового» продукту відсутня.

Тож необхідно використовувати рішення засновані на підборі рекомендацій на основі продукту, що розглядається на даний момент, рішеннях інших покупців та історії усіх покупок. Подібний функціонал можна знайти на сайті Gearbest. На рис. 1.4 представлені два елементи «Recommended for you» та «Customers also shopped for». Точний алгоритм підбору продуктів для першої секції невідомий, але можна зробити припущення, що вибір виконується на основі вже переглянутих

продуктів та популярних на даний момент. Наступна секція інформує про товари, які покупці обирають разом з даним.

The screenshot displays the Gearbest website interface. At the top, there is a search bar with 'samsung phones' entered. Below the search bar are navigation tabs: 'Category', 'NEW ARRIVALS', 'DEALS', 'LOCAL WAREHOUSE', and 'COMMUNITY'. The main product page features a large image of a box of blue surgical masks. To the right of the image, the product title is 'Disposable Isolation Face Mask Activated Carbon Anti-Dust Bacteria Proof Surgical Masks 50pcs - Deep Sky Blue'. Below the title, there is a 'Super Deals' banner with a price of \$29.99 (RRP: \$49.99, 40% OFF). The product is currently 'Out of Stock'. Below the product page, there are two recommendation sections: 'Recommended for You' and 'Customers also shopped for'. Each section contains a grid of product cards with images, titles, prices, and discount percentages.

Recommended for You

Product	Price	Discount
Disposable Isolation Medical Face Mask Ant...	\$29.99	29% OFF
Disposable Medical Mask 3 Ply Safety...	\$5.27	
LED Outdoor Camping Survival Whistle...	\$6.75	
JJFA JF- 606 Fine Stainless Steel Tweezers	\$4.64	
YG - 300 LCD Projector 320 x 240 Home Media...	\$37.99	17% OFF
Sport Silicone Watch Band Wrist Strap For...	\$5.55	
Jumbo Squishy Crystal Mud Plasticine Pressur...	\$6.92	12% OFF
Beautiful Cherry Wood Pattern Carpet Home...	\$5.98	20% OFF

Customers also shopped for

Product	Price	Discount
Face Protection Cap PVC Mask Protective Cap	\$8.69	
Europe Star Disposable Mask 50PCS	\$35.99	30% OFF
Lesbian Child Star Networks Red Mask...	\$3.02	22% OFF
5pcs KF94 Mask Dustproof Protective...	\$26.99	27% OFF
M002 3-Ply Ear Loop Adult Disposable...	\$9.64	
5PCS KN95 Masks with Breathing Valve Elastic...	\$10.99	23% OFF
5 Anti-stab Stab-resistant 5-level HPEE...	\$3.30	
100Pcs Disposable Medical Gloves Strong...	\$14.99	20% OFF

Рисунок 1.4 – Рекомендації до покупок на сайті Gearbest

Алгоритми вибору таких рекомендацій є досить примітивними, адже він базується виключно на факті того, які продукти були переглянуті/придбані раніше не беручи до уваги їх класифікацію, характеристики. Як наслідок, частина запропонованих продуктів не має жодного зв'язку з точки зору покупця і навряд чи зацікавлять його.

1.3 Постановка задачі

Необхідно спроектувати автоматизовану систему підтримки прийняття рішень для вибору товарів при здійсненні покупок в інтернет-магазині. Система має представляти собою набір програмних модулів, кожен з яких відповідатиме за окрему частину функціоналу, як то: мінімізація даних, обробка даних та представлення їх у виді, необхідним для подальшого аналізу, прогнозування та здійснення рекомендацій, API інтерфейс для комунікації з сервером веб-магазину представлена у вигляді REST-сервісу.

Дослідження, що проводяться в рамках атестаційної роботи мають базуватися на основі реальних даних, що якнайближче відображають можливі рішення, які приймають покупці в процесі здійснення покупок. Дані, що використовуються повинні відповідати певним вимогам, а саме:

- необхідна інформація, котра відобразатиме вхідні дані, які мав покупець на момент вибору товару для покупки та рішення, яке він в результаті прийняв, тобто товар, що в кінці був придбаний;
- кожен товар повинен володіти певним спектром характеристик, які користувач інтернет-магазину брав до уваги при здійсненні свого вибору;
- важливо розділяти причетність продуктів до певних категорій, адже при виборі товарів з різних категорій користувачі керуються різними мотивами і товари мають різний набір характеристик;
- кількість зібраних даних має бути якомога більшою задля досягнення найточніших результатів роботи СППР (Система підтримки прийняття рішень);

Група таких даних (набір прецедентів) є базою або ж основою для своєрідного навчання системи, вона допоможе їй сформувати образ певного канонічного покупця та отримати уявлення про його дії та рішення, що він приймає. Маючи набір необхідних даних, що відповідають переліченим вище вимогам система повинна приймати на вхід перелік товарів, котрі покупець вже

переглянув, та які вважаються перспективними для покупки і передати їх на розгляд абстрактного канонічного покупця, котрий базуючись на раніше здійснених покупках представить свій рекомендований список схожих товарів, що не являються частиною вхідних даних. Схематичне спрощене відображення принципу роботи з інформацією можна розглянути на рисунку 1.5.

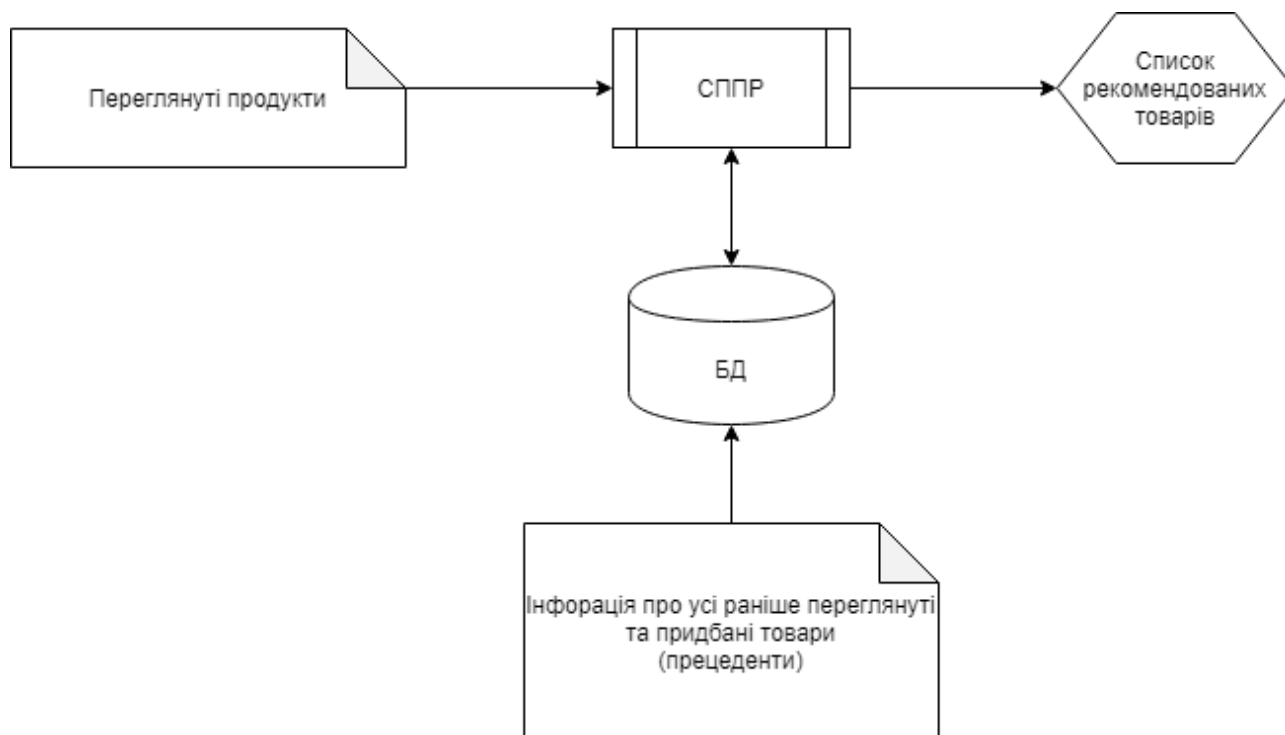


Рисунок 1.5 – Вхідна та вихідна інформація роботи програми

Однією з проблем, котрі розглядатимуться в ці роботі є існування неявних властивостей продуктів, на котрі покупець звертає увагу. До таких можна віднести, наприклад, зовнішній вигляд, взагалі будь-які суб'єктивні фактори здійснення вибору. Це є важливою причиною наявності похибок в СППР, адже переважна більшість таких систем не має чітких даних, або ж навіть не може зробити припущення щодо факторів, котрі впливають суб'єктивність рішень.

З часом потреби покупців змінюються, з'являються нові товари, або ж деякі пропозиції і зовсім зникають. В умовах постійних сезонних коливань та обов'язкових періодичних змін попиту та пропозиції в сфері електронної комерції

з'являється потреба в коректуванні роботи СППР[3], тобто її оновлення. Початкова база прецедентів може втратити свою актуальність чи не в повній мірі відобразити сучасні потреби в здійсненні покупок. Отже, варто постійно оновлювати базу прецедентів, що дозволить системі приймати найбільш актуальні рішення, котрі в повній мірі відобразатимуть сучасні тенденції.

Система працюватиме з потенційно великим об'ємом даних, тож дуже важливо, щоб сервер був якомога ефективніше спроектований та оптимально використовував обчислювані ресурси. Алгоритми, що будуть використовуватися повинні бути оптимізовані з врахуванням типу даних відповідно до обраної предметної області.

В загалом дії системи можна розділити на дві частини: аналіз прецедентів, прийняття рішення на основі вхідних даних. Треба звернути увагу на те, що кількість інформації для обробки на першому етапі буде досить масштабною і, очевидно, займатиме багато часу для обробки, а отже, варто звернути увагу на можливість багатопоточної обробки даних. У випадку з другим етапом очевидним є визначення потреби швидкодії, адже дана операція потенційно виконуватиметься велику кількість разів, можливо, навіть за запитом декількох користувачів одночасно. Отже, необхідно зробити упор на попередню підготовку даних, зберігання деякої структури попередньо оброблених та підготовлених одиниці інформації, котрі допоможуть швидко прийняти рішення.

Розроблений продукт повинен декларувати інтерфейс взаємодії, що буде використовуватися серверною частиною потенційного інтернет-магазину, як було раніше вирішено, він має представити набір REST ресурсів, що мають бути максимально простими та зрозумілими для використання, а їх назви повинні в повній мірі відобразити дії, що вони виконують.

Важливо зауважити, що подібна система не може гарантувати прийняття абсолютно правильного рішення[4], враховуючи розкид в даних, що використовуються як база прецедентів та суб'єктивності раніше прийнятих рішень покупцями інтернет-магазину.

2 ПЕРЕЛІК ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Кожна програмна система – це перетворювач, функцією якого є визначене оброблення даних і вивід отриманих результатів. З метою побудови програмної системи до неї, насамперед, формуються вимоги до умов виконання функції і обробки даних. У загальному випадку під вимогами до програмної системи розуміють властивості, які повинна мати система для виконання запропонованих замовником функцій. В рамках даної дипломної роботи необхідно розробити автоматизовану систему підтримки прийняття рішень, ціллю якої є допомога покупцям Інтернет-магазинів в виборі товарів при здійсненні покупок. За допомогою СППР може проводитись вибір рішень у певних неструктурованих і слабо структурованих задачах, у тому числі й тих, що мають багато критеріїв. Для створення якісного сервісу необхідно чітко визначитися з вимогами до нього.

2.1 Призначення розробки

Система підтримки прийняття рішень при здійсненні покупок в Інтернет-магазині покликана допомогти користувачу в виборі товарів. Подібна розробка буде корисною не лише для покупця, а й для власників магазинів, адже від дій та рішень відвідувачів напряду залежить їх дохід.

Реалізований програмний продукт має вирішити ряд проблем, помилок, до яких схильні покупці. Під час пошуку бажаного товару користувач може переглядати безліч потенційних варіантів керуючись певними вимогами, їх характеристиками. Звичайно, необхідно володіти деякими знаннями про товари обраної категорії щоб прийняти компетентне рішення. Відсутність таких знань може привести до вибору непідходящого товару, або ж до відмови в покупці в даному магазині зовсім. Цю проблему може вирішити розроблена система, шляхом

здійснення рекомендацій на основі рішень інших покупців. Існування подібних рекомендацій також допоможе в вирішенні деяких проблем викликаних людським фактором. Так, покупець може бути спантеличений великою кількістю наявних товарів, що характерно переважній більшості сучасних магазинів, як наслідок, пропустити найбільш підходящий товар, або ж неуважно вивчити опис певного товару, що досить ймовірно при виборі, наприклад, складної електроніки, котра має великий перелік характеристик.

Отож, отримуючи рекомендації при здійсненні покупок підвищується шанс позитивного досвіду покупця, що в свою чергу безпосередньо впливає на кількість придбаних товарів, задоволеність покупців та продажі в магазині, а це неминуче призводить до збільшення прибутку.

2.2 Вимоги до програмного продукту

Розглянемо загальні вимоги до програмної системи. При виборі мови програмування треба керуватися наявністю широкої общини розробників, що її використовують, це спростить вирішення проблем, котрі неодмінно виникнуть в процесі роботи. Також обрана мова програмування повинна супроводжуватися широкими засобами для роботи з WEB-технологіями та базами даних. Бажаним є наявністю хоча б базових навичок в її використанні. Таким вимогам відповідає мова Java[5].

Серед допоміжних засобів варто звернути увагу на рішення для управління залежностями, що дозволить легко підтримувати модульність та, як наслідок, масштабованість системи, наприклад, Spring Framework[6].

Для розробки RESTful[7] архітектури та представлення інтерфейсу взаємодії на базі HTTP протоколу варто використати модуль раніше представленого фреймворку, а саме – SpringMVC[8]. Дані засоби дозволять спроектувати багат шарову архітектуру, в якій роль точок взаємодії з користувачем буде

відведена REST контролерам, що дозволяє отримувати та відправляти різноформатні дані по протоколу HTTP.

Враховуючи специфіку роботи системи та інформація, котрою вона оперує, необхідно обрати засоби для керування даними, тобто СУБД. Найбільш поширеними є реляційні СУБД, котрі добре працюють з різними типами, групами даних, що мають певні зв'язки між собою, вони ж, як правило, володіють потужними засобами для обробки інформації, що зберігається. Проте найголовнішою вимогою при виборі має стати швидкодія, адже об'єм потенційної інформації буде досить великим, історія дій користувачів Інтернет-магазинів при здійсненні покупок може сягати мільйонів, або, навіть, мільярдів переважно однотипних записів, без складних зв'язків. В такому випадку треба зупинитися на NoSQL базах даних, наприклад, MongoDB[9].

Корисними засобами, що спростять процес розробки стануть система автоматизованої роботи з програмними проектами Apache Maven та система контролю версій Git.

Якщо розглядати програмну систему в розрізі функціоналу, то варто виділити декілька ключових моментів, котрих потрібно дотримуватися. Система передбачає роботу з великим об'ємом даних, отриманих в результаті роботи певного магазину. Очевидно, що отримана початкова інформація може бути представлена в неоптимальному вигляді, отже, необхідно виконати певну попередню обробку даних, мінімізувати їх аби далі представити в найбільш зручним для роботи вигляді.

Маючи попередньо підготовлений набір прецедентів система повинна прийняти рекомендаційне рішення щодо здійснення покупки, при цьому необхідно звернути увагу на категоризацію продуктів. Враховуючи відсутність можливості прийняти абсолютно правильне рішення, система повинна представити перелік можливих рішень з урахуванням їх ймовірності.

Для отримання вхідних та передачі вихідних даних має існувати певний інтерфейс. Важливо, щоб він був якомога простішим та інтуїтивно зрозумілим, при

цьому неодмінним є підтримка багатозадачності, тобто можливості опрацьовувати декілька запитів одночасно.

Отже, у ході розробки програмної системи необхідно проаналізувати предметну область електронної комерції, методів підтримки прийняття рішень, спроектувати та розробити за допомогою обраних технологій та мови програмування автоматизовану систему, котра здійснюватиме рекомендації щодо вибору продуктів для покупки.

3 ОПИС МЕТОДІВ, МОДЕЛЕЙ ТА АЛГОРИТМІВ

3.1 Аналіз та UML-моделювання предметної області

Для правильної розробки системи, котра буде рекомендувати товари для покупки було проаналізовано предметну область, розглянуто різні типи Інтернет-магазинів, види товарів та послуг, що вони представляють та їх аудиторію, тобто покупців. А також сценарії дій покупців під час здійснення покупок. Ця інформація дозволить в повній мірі оцінити усі процеси, що відбуваються при виборі товарів та дозволить розробити компетентну систему, котра відповідатиме вимогам сучасних магазинів та їх відвідувачів.

Так магазини можна розділити на дві основні категорії за способом продажу:

- фіксована ціна товару - з доставкою, включеної у вартість, або з доставкою, яка вважається окремо, після оформлення замовлення (дуже часто ефект низької ціни буває зіпсований через високу вартість доставки, продавці навмисне можуть ставити низьку вартість на товар, а на доставку навпаки - високу, на чому і заробляють).

- система аукціону - на товар оголошується аукціон. Крім початкової ціни, продавець може оголошувати так звану блиц-ціну - це вартість, за яку продавець готовий віддати товар без торгу.

Було вирішено сфокусуватися на магазинах першого типу, адже, як правило, вони є більш поширеними, та дійсно потребують існування деякого консультанта, на відміну від аукціонів, де покупці вже чітко знають, що вони бажають придбати та кожен товар, як правило, є одиничною пропозицією.

Переглянувши декілька існуючих Інтернет-магазинів можна сформулювати основні принципи здійснення покупки. Кожен сайт містить певний каталог товарів і послуг (існують магазини одного товару, в цьому випадку каталог відсутній і такі сайти не є цільовими). Кожна одиниця номенклатури товару або послуги складається з артикулу, найменування, ціни та набору деяких характеристик,

особливих для певної категорії товарів, що також іноді супроводжується зображенням. Електронні товари в каталозі можуть бути представлені попередніми копіями (наприклад, зменшені зображення, звукові файли погіршеної якості, фрагменти відео, демо-версії програмного забезпечення), що не мають комерційної привабливості.

Вивчаючи представлені у мережі веб-сайтів електронної комерції було виділено основні види товарів, що реалізуються:

- велика і дрібна побутова техніка;
- одяг та взуття;
- комп'ютерна техніка та інші електронні гаджети;
- товари для дітей (іграшки, одяг, взуття, книги, текстиль, аксесуари);
- запчастини для автомобілів і техніки;
- аксесуари, шкiргалантерея, ювелiрнi вироби;
- побутова хiмiя, товари для дому, посуд;
- канцтовари для офiсу та школи;
- сантехнiка;
- спортивнi товари, велосипеди та аксесуари до нас;
- товари для риболовлi, туризму i подорожей;
- квіткова продукцiя i сувенiри;
- лiки i БАДи;
- продукти харчування, чай, кава та iн.

Подiбне рiзноманiття видiв товарiв та рiзношерстiсть набору характеристик, котрими вони володiють визначає одну з основних вимог до програмного продукту – унiверсальнiсть. Очевидно, що гарантуючи можливiсть роботи системи в магазинах, котрi представляють такий широкий спектр товарiв та послуг, постає проблема неможливостi здiйснення експертної оцiнки та визначення важливостi окремих характеристик в наборi наявних для товарiв певної категорiї. Тобто критерiї, що будуть розглядатися СППР є неоднорiдними, iх набiр є унiкальним для кожної групи, абсолютно спiльним є лише наявнiсть цiни.

Спробуємо чітко виділити суть предметної області. Взагалі сутність – це будь-який конкретний або абстрактний об'єкт в аналізованій предметній області. Також можна сказати що це базові типи інформації, які зберігаються в базі даних. До сутностей можуть відноситися: клієнти, товари і т.д. При складанні списку сутностей потрібно постаратися виділити усі можливі, проаналізувати їх і видалити повторювані.

Самими очевидними базовими сутностями предметної області, що розглядається, є покупець та товар, котрій він обирає, кошик. Хоча це й становить основу будь-якого Інтернет-магазину, метою роботи є розробка системи підтримки прийняття рішень, а не реалізація магазину. Поставлена задача зумовлює необхідність роботи з дещо іншими сутностями, котрі тим не менш є похідними від вище зазначених.

Так, сам факт існування покупця або товару не несе ніякої корисної інформації для системи. В даному випадку головним є випадок перегляду товару та додання його до кошика. Тобто головна сутність – це подія, що відбулась за участі покупця щодо певного товару і система, в свою чергу, має розрахувати ймовірність такої події для інших продуктів і представити найбільш ймовірні з них кінцевому користувачу.

Очевидно, що покупець робить свій вибір на основі певних характеристик товарів, вони можуть бути спільними для деяких груп продуктів, але в більшій мірі така схожість властива товарам в одній категорії. Варто звернути увагу на те, що характеристики можуть змінюватися з часом, що безпосередньо впливає на вибір в покупці. Отже, ще двома сутностями системи є характеристики (атрибути) товарів та їх категорії.

Таким чином подія (перегляду, переміщення в кошик товару), атрибут товару та категорія є основними сутностями для системи, хоча, очевидно, що при розробці може виникнути потреба в використанні деяких допоміжних сутностей.

Проаналізувавши предметну галузь та виділивши основні сутності в ній можна отримати базове уявлення про те, як має працювати система. Схематично

основні етапи роботи з СППР можна розглянути на рисунку 3.1. Представлена діаграма прецедентів чітко виділяє існування двох акторів: покупця та серверу Інтернет-магазину, який використовує СППР.

Покупець здійснює звичні дії перегляду продуктів та додає обрані до кошика. Інформацію про такі події реєструє Інтернет-магазин та згодом відправляє до СППР, ці данні становитимуть основу на базі якої система буде «навчатися». Переславши певний набір даних магазин має ініціювати процес аналізу представлених подій, після закінчення якого, програмний продукт буде в повній мірі готовий для прийняття рішення щодо рекомендації товарів для покупки на базі вже переглянутих продуктів потенційним покупцем.



Рисунок 3.1 – Діаграма прецедентів

Отже, в результаті аналізу предметної галузі, а саме огляду Інтернет-магазинів, видів товарів та послуг, котрі вони представляють були обрані фактори які треба враховувати при побудові оптимальної автоматизованої системи підтримки прийняття рішень при здійсненні покупок в Інтернет-магазині.

3.2 Аналіз існуючих методів підтримки прийняття рішень

Система підтримки прийняття рішень (СППР) - це інформаційна система, яка підтримує бізнес або організаційну діяльність щодо прийняття рішень. СППР обслуговують рівні управління, операції та планування організації (як правило, середнього та вищого менеджменту) та допомагають людям приймати рішення щодо задач, які можуть швидко змінюватися і не можуть бути легко визначені заздалегідь, тобто неструктуровані та напівструктуровані проблеми прийняття рішення. Системи підтримки прийняття рішень можуть бути як повністю комп'ютеризовані, так і на базі людей, або комбінацією обох.

СППР включають системи, засновані на знаннях. Правильно розроблена СППР - це інтерактивна система, заснована на програмному забезпеченні, яка покликана допомогти тим, хто приймає рішення, зібрати корисну інформацію з комбінації необроблених даних, документів та особистих знань або ділових моделей для виявлення та вирішення проблем та прийняття рішень.

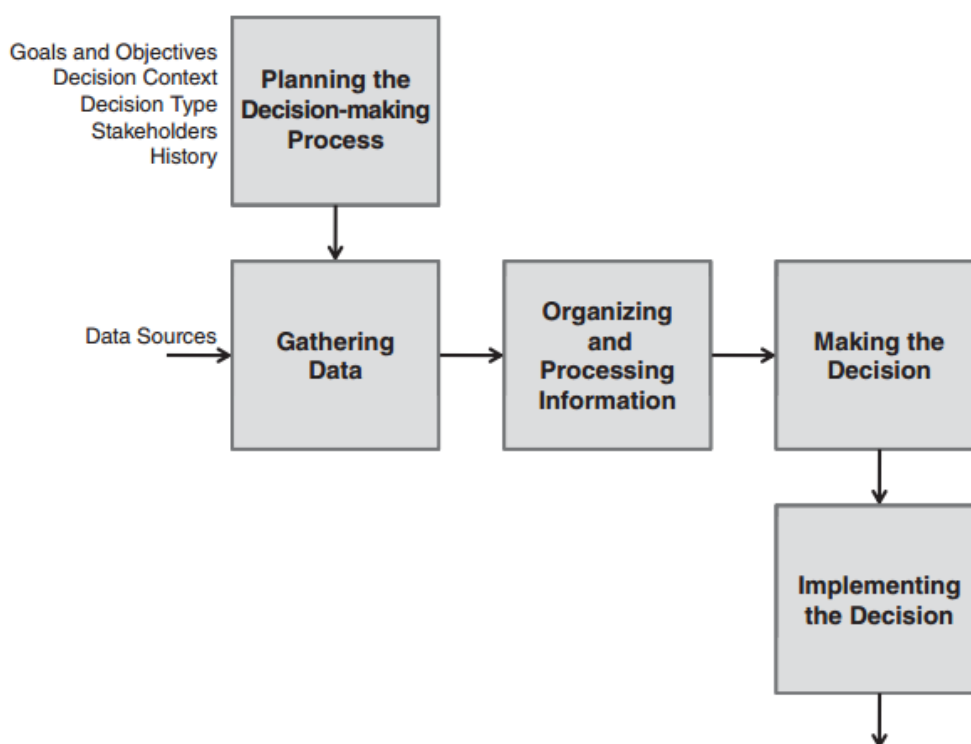


Рисунок 3.2 – Стадії процесу прийняття рішень

Підтримка прийняття рішень за допомогою інформаційних технологій, включаючи аналіз і вироблення альтернатив, в СППР може здійснюватися наступними методами:

- інформаційний пошук;
- інтелектуальний аналіз даних;
- витяг (пошук) знань в базах даних;
- міркування на основі прецедентів;
- імітаційне моделювання;
- генетичні алгоритми;
- штучні нейронні мережі;
- штучний інтелект.

В СППР можуть використовуватися суб'єктивні судження користувача або експерта.

Інтелектуальний аналіз даних (data mining) - це виявлення прихованих закономірностей або взаємозв'язків між змінними у великих масивах необроблених даних. Термін "data mining" введений Г. Пятецький-Шапіро в 1989 р. Такий аналіз передбачає вирішення завдань класифікації, моделювання і прогнозування, кластеризації, скорочення опису, асоціації, аналізу відхилень, візуалізації і ін.

Кластерний аналіз (Data clustering) - задача розбиття заданої вибірки об'єктів (ситуацій) на підмножини, які називаються кластерами, так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися. Завдання кластеризації відноситься до широкого класу задач навчання без учителя.

Дуже часто дані, з якими стикається технологія Data Mining, мають такі важливі особливості:

- висока розмірність (тисячі полів) і великий обсяг (сотні тисяч і мільйони записів) таблиць баз даних і сховищ даних (надвеликі бази даних);
- набори даних містять велику кількість числових і категорійних атрибутів.

Всі атрибути або ознаки об'єктів діляться на числові (numerical) і категорійні (categorical). Числові атрибути - це такі, які можуть бути впорядковані в просторі, відповідно категорійні - яке не можуть бути впорядковані. Наприклад, атрибут «вік» - числовий, а «колір» - категорійний.

Більшість алгоритмів кластеризації припускають порівняння об'єктів між собою на основі певної міри близькості (подібності). Мірою близькості називається величина, що має межу і зростаюча зі збільшенням близькості об'єктів. Міри схожості «винаходяться» за спеціальними правилами, а вибір конкретних заходів залежить від завдання, а також від шкали вимірювань. В якості запобіжного близькості для числових атрибутів дуже часто використовується евклідова відстань, що обчислюється за формулою:

$$D(x, y) = \sqrt{\sum_i (x - y)^2}$$

Одним з найбільш ефективних та популярних алгоритмів кластеризації є DBSCAN – це алгоритм просторової кластеризації на основі щільності з урахуванням шумів. Цей алгоритм кластеризації, заснований на щільності - якщо дано набір точок у деякому просторі, алгоритм групує разом тісно розміщені пункти, помічаючи як викиди точки, які знаходяться самотньо в областях з малою щільністю. Характерною особливістю цього алгоритму є відсутність необхідності попереднього визначення кількості кластерів (чим відомий алгоритм k-means). Даний алгоритм кластеризації можна застосувати для групування отриманих подій перегляду продуктів користувачем в рамках однієї сесії.

Міркування на основі прецедентів (case-based reasoning) - ще один метод підтримки прийняття рішень за допомогою інформаційних технологій. Прецедент - випадок, що мав місце буди раніше і що служить прикладом або виправданням для наступних випадків подібного роду. Висновок на основі прецедентів є підходом, що дозволяє вирішити нове завдання, використовуючи або адаптуючи

рішення вже відомої задачі. Як правило, такі методи міркувань передбачають чотири основні етапи, що утворюють так званий цикл міркування на основі прецедентів, або CBR-цикл. Існує цілий ряд методів вилучення прецедентів і їх модифікацій:

- метод найближчого сусіда (nearest neighbor);
- метод вилучення прецедентів на основі дерев рішень;
- метод вилучення прецедентів на основі знань;
- метод вилучення з урахуванням застосування прецедентів.

Крім розглянутих методів для отримання прецедентів можуть успішно застосовуватися і інші методи, наприклад, апарат штучних нейронних мереж.

В СППР можуть використовуватися один або кілька методів, за допомогою яких будуть відбиратися критерії вибору оптимального рішення. Вони повинні працювати як з кількісними, так і з якісними значеннями критеріїв.

Ступінь досягнення мети при прийнятті рішень оцінюється на основі критеріїв, тому їх відбір, їх якість, їх кількість - дуже важливий процес, переоцінити який складно. Саме тому їх називають критеріями вибору оптимального рішення або критеріями ефективності рішення. Якщо помилитися з вибором критеріїв, то альтернативи не будуть оцінені належним чином і прийняття рішення не буде найкращим з можливих.

Критерії можуть бути як об'єктивними, так і суб'єктивними. Критерії, які не можуть бути виміряні об'єктивно, будуть суб'єктивними: наприклад, імідж компанії, якість продукту, його зовнішній вигляд можуть бути виміряні лише суб'єктивно. Формування оціночної системи критеріїв - найважливіший крок в прийнятті рішень. Проте часом система може супроводжуватися великою кількістю різноманітних критеріїв та відсутністю знань про їх природу, важливість та цінність, що робить неможливим виділення окремих критеріїв та присвоєння певних вагових коефіцієнтів. Такі критерії можуть представляти собою як прості числа чи слова так і їх набір, наприклад, опис товару, в такому випадку особливу увагу треба приділити визначення ступеня схожості описів декількох

альтернатив[10]. До того ж можуть існувати деякі неявні критерії, котрі не можна знайти навіть шляхом факторного аналізу.

3.3 Математичне моделювання

Основними (базовими) сутностями предметної галузі в області інтернет-магазинів, а отже, і математичної моделі розроблюваної системи є подія перегляду продукту E_v та його вибору (додання до кошика) E_c , категорійна структура продуктів C та атрибут продукту A . Отже, розглянемо докладніше їх структуру та характеристики та промодельюємо предметну галузь.

Задача пошуку оптимальних продуктів для здійснення рекомендацій в прийнятті рішень (здійсненні покупки):

$$X = \langle P_v, Y, D \rangle,$$

де P_v – нещодавно переглянуті продукти покупцем, який потребує підтримки в прийнятті рішень

D – рішення прийняте системою (рекомендовані продукти),

Y – набір попередньо проаналізованих прецедентів на базі яких приймається рішення.

Насамперед для забезпечення можливості функціонування системи необхідно підготувати базу прецедентів, котра повинна бути сформована на основі первинних даних представлених Інтернет-магазином[11]. Таким чином, потрібно володіти інформацією про події перегляду продуктів E_v та події їх переміщення до кошика E_c . Але варто зазначити, що ці події залежать від часу та відносяться до певного користувача. За увесь час взаємодії с сайтом один користувач може

здійснювати декілька покупок з певним інтервалом, такі покупки є незалежними, а отже, виникає потреба визначення сесійності для відвідувачів Інтернет-магазину:

$$S = S(t) = \langle E_v, E_c, V \rangle,$$

де V – покупець магазину.

При цьому кожен набір подій також залежить від часу:

$$E_v = E_v(t), E_c = E_c(t).$$

Сукупність цих подій можна описати наступним чином:

$$E_v = \{e_{v_i}(t)\}_{i=1}^n, \text{ та } E_c = \{e_{c_i}(t)\}_{i=1}^m,$$

де n та m – кількість відповідних подій, при цьому $n \geq m$ і $n > 0, m \geq 0$, тобто кількість придбаних продуктів не може перевищувати кількість переглянутих.

В ролі критеріїв моделі, що розглядається, виступають атрибути товарів (характеристики). Кожен товар володіє чітко визначеним набором властивостей, котрі тим не менш можуть змінюватися з плином часом, що пояснюється простим редагуванням опису продуктів. Отже, множину атрибутів товарів можна описати таким способом:

$$A = A(t) = \{a_i\}_{i=1}^n,$$

де n – кількість критеріїв, що може змінюватися в залежності від категорії, до якої належить продукт.

При наявності інформації щодо минулих сесій, критеріїв та категоризаційного розподілення продуктів маємо можливість сформулювати математичне визначення підготовленого прецеденту:

$$Y = \langle S, A, C \rangle.$$

Ця сутність в повній мірі відображає рішення прийняті реальними покупцями раніше, на основі яких здійснюється «навчання» системи[12]. Володіючи усіма необхідними даними система може прийняти рішення, рекомендувавши перелік продуктів, що можна представити у вигляді:

$$D = \langle P_r, L \rangle,$$

при цьому $P_r = \{p_{r_i}\}_{i=1}^n$ – продукти рекомендовані для купівлі,

L – ймовірність правильності вибору певного продукту за думкою системи.

3.4 Розробка схеми бази даних для зберігання математичної моделі

Зберігання математичної моделі неможливо без додаткових сутностей, які б описували концептуальну область електронної комерції. В нашому випадку є сім сутностей: продукт, подія, сесія, категорія, характеристика продукту, прецедент, рішення. Зв'язки між цими сутностями показані на концептуальній моделі, яка наведена на рисунку 3.3. Концептуальна модель складається з переліку взаємопов'язаних понять, які використовуються з метою описання певної предметної області, разом з властивостями та характеристиками, класифікацією цих понять за типами, ситуаціями, ознаками в даній області і законів протікання процесів.

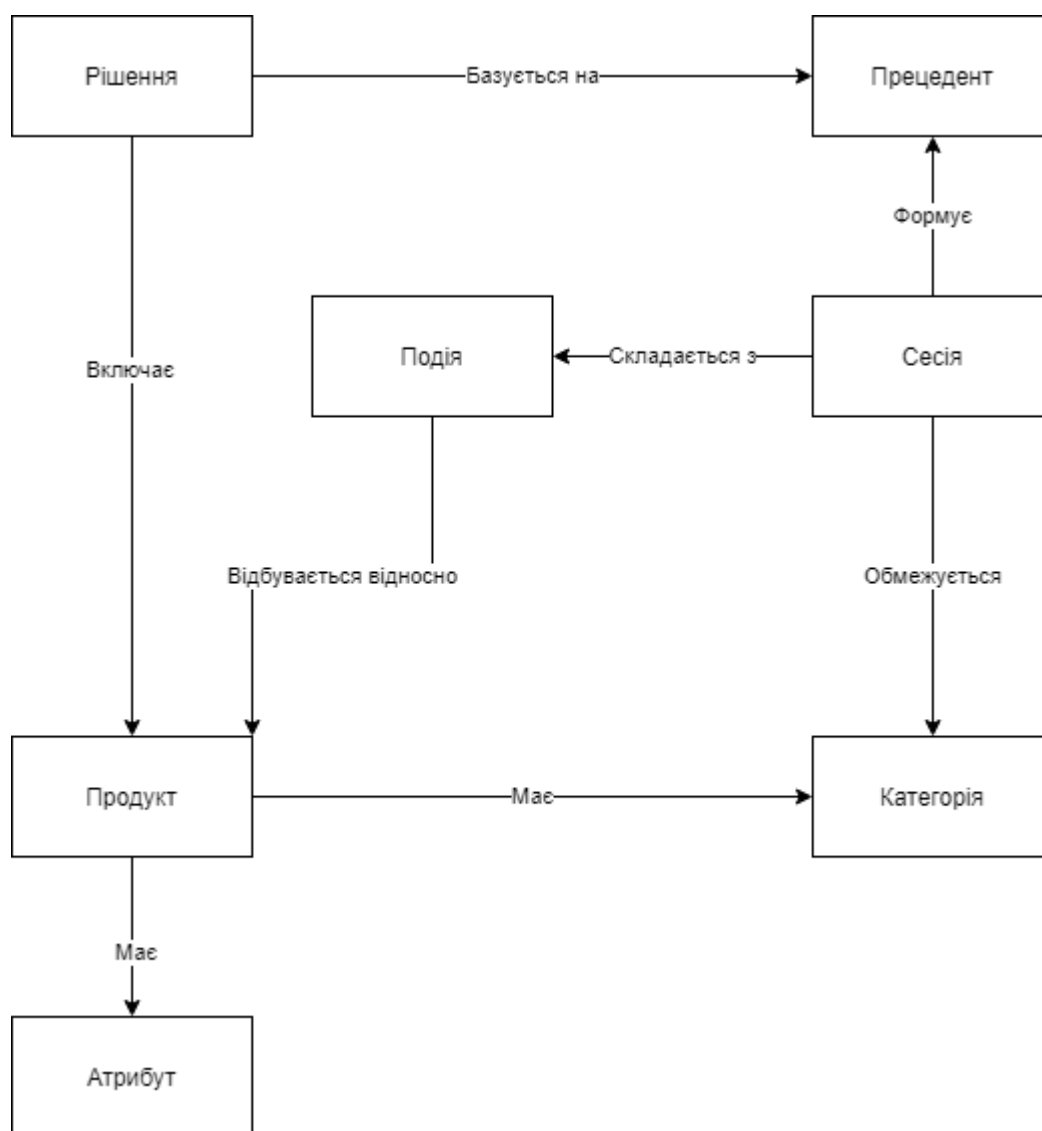


Рисунок 3.3 – Концептуальна модель предметної області

Наведемо опис призначення кожної сутності у системі:

- рішення – результат роботи СППР, що несе в собі перелік продуктів та базується на проаналізованих прецедентах[13];
- прецедент – це сутність, що характеризує проблему вибору продукту, котра раніше мала місце бути, для якої відомі набір критеріїв та рішення, яке було в кінці прийнято;
- сесія – сутність агрегації подій за ознаками належності товарів до певної категорії, суб'єкта сесії (покупця) та часу, в рамках якого виконувались певні дії на сайті магазину;

- подія – дія, котра була здійснена покупцем в Інтернет-магазині відносно якогось продукту (перегляд, переміщення до кошика);
- товар – основний об’єкт, котрим оперує Інтернет-магазин, відноситься до певної категорії та має визначений набір характеристик;
- категорія – ознака класифікації товарів;
- атрибут – властивість продукту, що може змінюватися з часом, критерій для прийняття рішень.

Очевидно, що деякі з представлених вище сутностей є проміжними та необхідні лиш на деяких етапах роботи системи. Це зумовлює зменшення кількості даних, які треба зберігати в базі даних. Так, можна відмовитися від зберігання сесій (є допоміжною для формування прецедентів) та продуктів (вхідний параметр для системи). Беручи до уваги вказані спрощення та потенційно великий об’єм даних для зберігання, звернемося для нереляційних NoSQL баз даних.

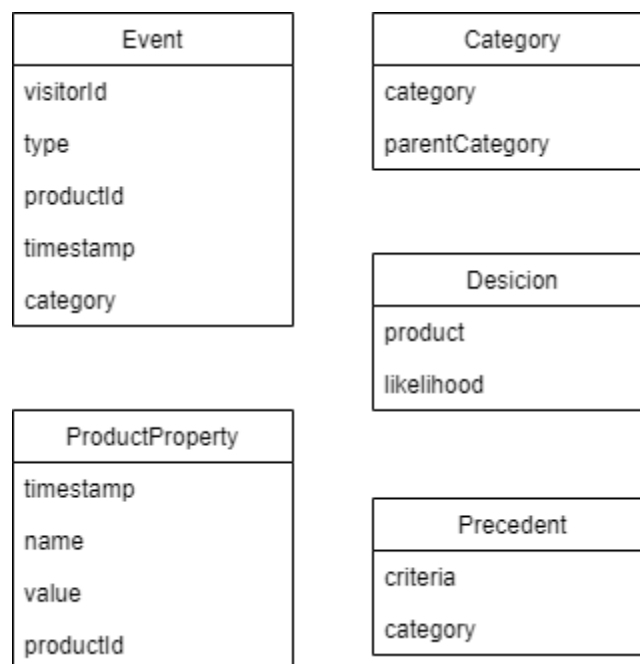


Рисунок 3.4 – Схема бази даних

Загальна схема бази даних 5 колекцій (таблиць), що є достатнім для зберігання інформації про події, ієрархії категорій, атрибутів товарів, прецедентів

та результатів роботи системи, тобто прийнятих рішень. На схемі не має жодних зв'язків, що пояснюється відсутністю необхідності в їх існуванні та принципом роботи NoSQL баз даних взагалі.

3.5 Розробка алгоритму планування

Після проведеного аналізу методів планування було визначено, що для системи підтримки прийняття рішень буде використовуватися підхід міркувань на основі прецедентів.

Для початку дамо чітке визначення прецеденту в контексті даної роботи. Прецедент – це випадок, що мав місце раніше і служить прикладом або виправданням для наступних випадків подібного роду. Кожен прецедент повинен включати в себе інформацію щодо: початкової вибірки альтернатив, альтернативи обраної експертом, переліком критеріїв, якими керувався експерт при здійсненні вибору. Спроектуювши ці вимоги на предметну область, що розглядається в даній роботі, конкретизуємо їх як: події перегляду товарів, вибір товарів для покупки та характеристики товарів відповідно.

Алгоритм роботи кожної СППР можна узагальнено розділити на такі етапи:

- збір інформації
- перетворення інформації до виду готового для обробки
- аналіз попередньо підготовлених даних
- прийняття рішення

Розглянемо кожен з цих етапів в окремоті.

Насамперед початковий набір інформації для роботи системи повинен бути надісланий їй безпосереднім користувачем, тобто сервером Інтернет-магазину. Сайт повинен відслідковувати події перегляду та переміщення до кошика товарів, а також зберігати інформацію про продукти, їх характеристики і, використовуючи

представлений REST API інтерфейс, передати ці дані до системи. Безперечно, це найпростіша частина роботи то ж не будемо акцентувати особливої уваги на цьому.

Інформація, отримана системою, представлена у вигляді не готовому для аналізу, а, отже, потребує певних перетворень (рис. 3.5). Проблема полягає в розрізненості даних, усі події треба класифікувати за їх типом, належності до відвідувача, котрий їх здійснював та об'єкту події (продукту). У свою чергу, атрибути продуктів треба класифікувати за належністю до продукту, адже початкові дані не мають подібної структуризації.

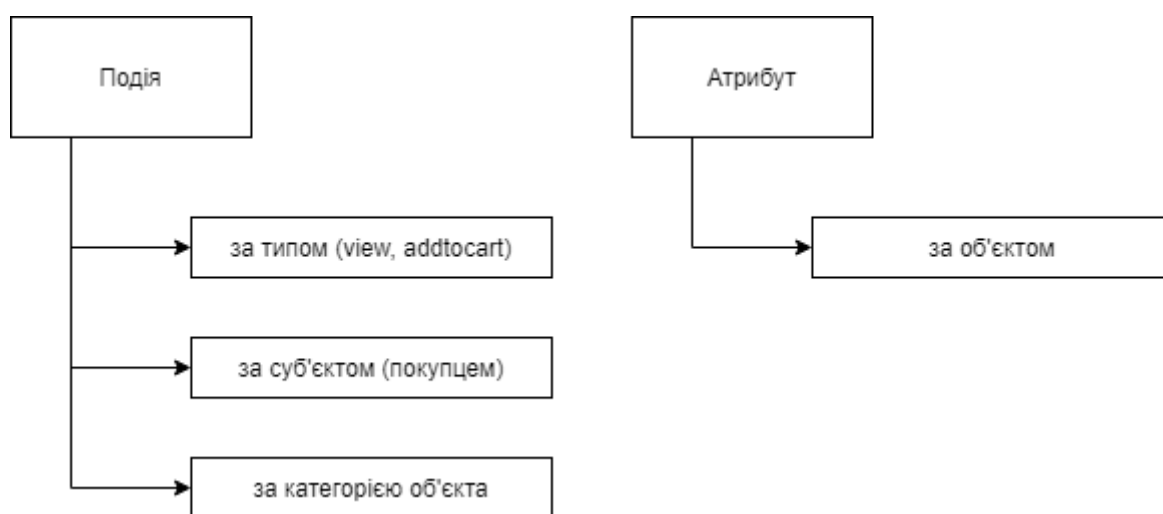


Рисунок 3.5 – Класифікація даних

Після виконання цієї попередньої обробки інформації можна сказати, що отриманні дані готові для подальшого аналізу.

Покупець може здійснювати декілька незалежних відвідувань сайту магазину для здійснення незалежних покупок, проте наявні дані не мають явних властивостей за якими можна визначити належність будь-якої події до окремої групи покупок. Так постає проблема групування даних за ознакою сесійності. Подібну задачу можна вирішити шляхом кластеризації спираючись на відомий час реєстрації події. Характерним є відсутність будь-яких відомостей щодо кількості необхідних даних. Було вирішено зупинити увагу на алгоритмі DBSCAN.

Розглянемо множину точок в деякому просторі, яка будемо кластеризувати. Метою кластеризації процедурою DBSCAN є поділ точок на ядрові, (щільно-)досяжні та викиди, які визначаються так:

- Точка p є ядровою, як хоча б $minPts$ знаходяться на відстані ϵ (відстань ϵ є радіусом окола p) від неї, включно з p . Кажуть, що ці точки безпосередньо досяжні з p .
- Точка q безпосередньо досяжна з p , якщо точка q знаходиться на відстані не більшій ніж ϵ від точки ядрової точки p .
- Точка q є досяжною з p , якщо існує шлях p_1, \dots, p_n з точок $p_1 = p$ та $p_n = q$, де кожна p_{i+1} безпосередньо досяжна з p_i (всі точки шляху повинні бути ядровими, можливо, за виключенням q).
- Всі точки не досяжні з будь-якої іншої точки є викидами.

Отже, ядрова точка p утворює «кластер» разом з усіма точками (не важливо ядрові чи ні) досяжними з неї. Кожен кластер містить принаймні одну ядрову точку. Не-ядрові точки можуть бути частиною кластера, тоді вони утворюють «ребро» кластер[14]а, оскільки з них не будуть досяжні інші точки.

Досяжність не є симетричним відношенням, бо, за визначенням, немає точок досяжних з не-ядрової точки (не-ядрова точка може бути досяжною, але відносно неї немає досяжних точок). Тому, поняття зв'язності потрібне для формального визначення степені кластеризації в DBSCAN. Дві точки p та q є щільно-зв'язними, якщо існує точка o з якої вони є досяжними. Щільно-зв'язність є симетричною.

Тоді кластер має дві властивості:

- всі точки кластера взаємно щільно-зв'язні;
- якщо точка є щільно-зв'язною з будь-якою точкою кластера, то вона так само належить кластеру.

На основі класифікованих та кластеризованих подій можемо сформулювати сутності сесії.

Наступним кроком є проведення багатокритеріального аналізу. Вважатимемо відомою множину аналогів, тобто товарів, які підлягають аналізу $S =$

$\{s_1, s_2, \dots, s_n\}$, тоді $C = \{c_1, c_2, \dots, c_m\}$ – множина кількісних та якісних критеріїв, за участі яких здійснюється оцінка альтернатив. Як правило, задача полягає в упорядкуванні елементів множини S по критеріям з множини C . При цьому для забезпечення можливості порівняння критеріїв зазвичай якісні критерії представляються у вигляді кількісних. Проте для здійснення подібних перетворень необхідно розуміти природу кожного критерію та мати деякого експерта, за рішенням якого назначатимуться альтернативні кількісні оцінки. Нажаль, в даній системі ці умови не виконуються, так кожен нечисловий атрибут (слово або їх набір) представлений у вигляді результату обробки хеш-функції слів, і назва атрибутів є невідомою.

Таким чином постає проблема критеріального аналізу в умовах неможливості перетворення вектору окремих критеріїв в скалярний інтегральний критерій.

Нехай $\mu^j(s_i)$ – число в діапазоні $[0, 1]$, котре характеризує рівень оцінки варіанта $s_i \in S$ по критерію $c_j \in C$: чим більше число $\mu^j(s_i)$, тим вище оцінка варіанта по критерію $c_j \in C, i = \overline{1, n}, j = \overline{1, m}$. Тоді критерій $c_j \in C$ можна представити у виді нечіткої множини \tilde{c}_j , котра задана на універсальній множині таким чином:

$$\tilde{c}_j = \left\{ \frac{\mu^j(s_1)}{s_1}, \frac{\mu^j(s_2)}{s_2}, \dots, \frac{\mu^j(s_n)}{s_n} \right\}$$

де $\mu^j(s_i)$ – ступінь належності елемента s_i до нечіткої множини \tilde{c}_j .

Виходячи з вищевказаного будемо формувати свого роду агрегатний продукт (прецедент, набір атрибутів/критеріїв, котрі відображають найліпше можливе рішення для системи), котрий об'єднує значення атрибутів альтернатив, при цьому кількісні атрибути об'єднуватимемо шляхом знаходження середнього арифметичного значення, а для складних якісних (текст) – формуватимемо мапу кількості входжень слів. При такому підході для визначення найліпшої альтернативи для числових значень буде знаходитися модуль відхилення від

канонічного агрегованого значення, а для складних текстових – визначення схожості значень шляхом пошуку входжень спільних слів. Чим менше модуль різниці числових значень та більша схожість тексту, тим ліпшим вибором є альтернатива, що розглядається. Тобто система передбачає визначення оцінки ефективності кожної альтернативи, а не лише вибір однієї єдиної[15].

У випадках, коли в множині прецедентних альтернатив спостерігаються значний розкид в значеннях окремих атрибутів можна говорити малу значимість такого критерію, а отже, усі подібні підлягають видаленню. Це дозволить зменшити об'єм інформації, що обробляється, та послабити вплив маловажливих критерій на прийняття рішення[16].

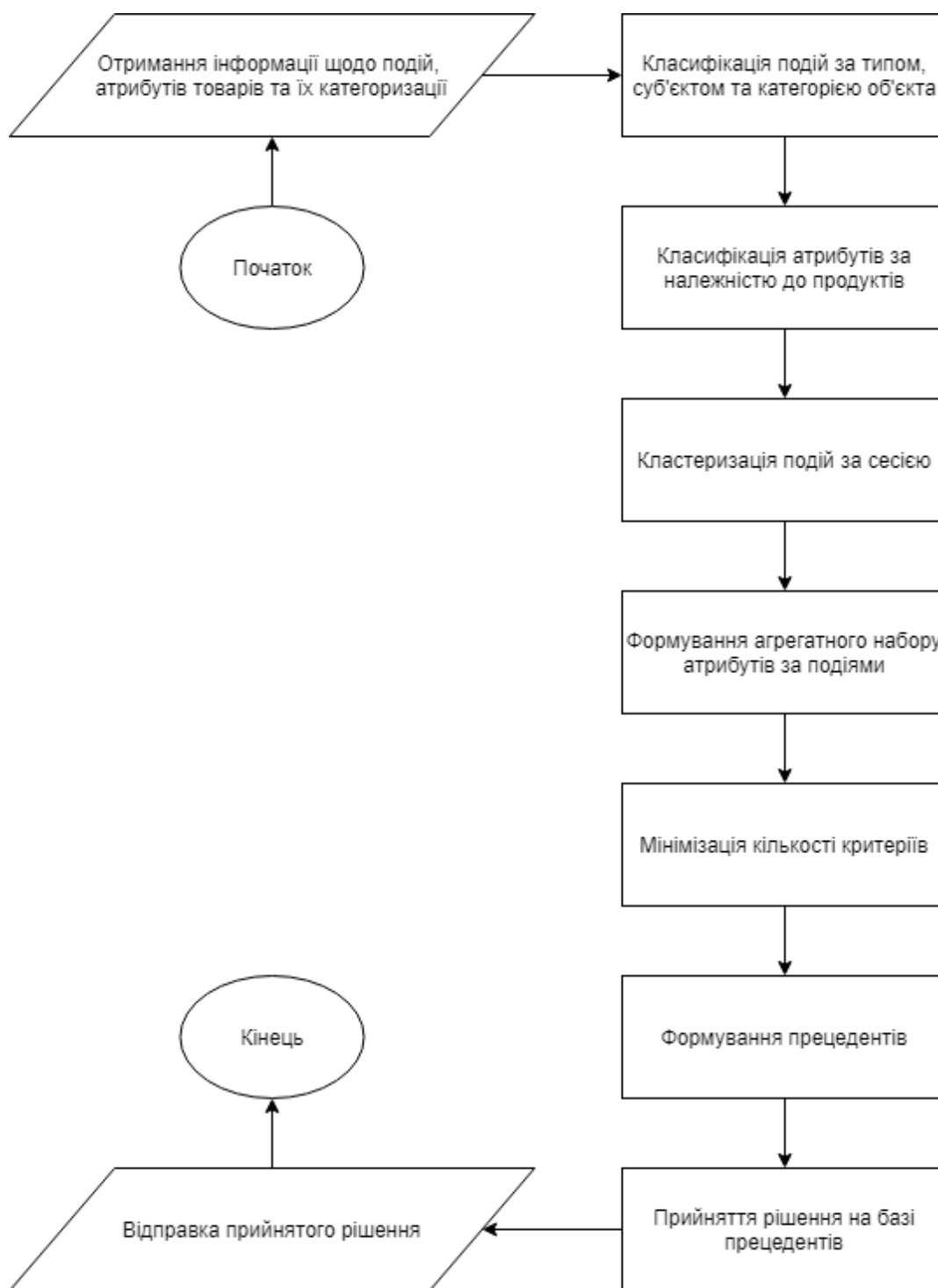


Рисунок 3.6 – Схема алгоритму роботи системи

На рисунку 3.6 представлена загальна схема алгоритму роботи системи без урахування особливостей кожного етапу в окремоті

3.6 Розробка архітектури застосунку

Перед розробкою програмного забезпечення необхідно було вирішити питання вибору типу архітектури та її проектування. На сьогоднішній день існує безліч архітектурних моделей та стилів, кожний з яких має свої переваги та недоліки. Навряд чи зараз можна зустріти систему розроблену виключно згідно єдиної архітектурної моделі. Більшість програмних продуктів представляють собою результат одночасного використання декількох стилей з певними модифікаціями. Це можна пояснити відсутністю ідеальної універсальної моделі, яка підходила би для вирішення певної проблеми.

Багаторівнева архітектура (також відома як n-рівнева архітектура)[17] - це клієнт-серверна архітектура, у якій відображення, обробка даних та їх менеджмент представляють фізично відокремлені частини. Найбільш поширеним використанням багаторівневої архітектури є трьохрівнева архітектура.

Він не бере участі в обробці інформації, а лише передає її, для заповнення в JSP, або, якщо це REST контролер, - клієнту в форматі JSON.

Рівень фасадів було вирішено ввести задля представлення прошарку між двома типами контролерів та сервісами. Він забезпечує універсальні методи, які згодом можуть бути використані для обробки інформації як для адміністративної частини так і для клієнтського додатку і, в той же час, дозволяє використовувати невелику кількість точок доступу, не звертаючи уваги на складну схему виклику сервісів.

Рівень сервісів - головна частина системи, що містить бізнес логіку. Саме на цьому рівні відбувається безпосередня обробка інформації.

Рівень даних забезпечує можливість взаємодії з базою даних для редагування даних в ній інформації. Це можливо завдяки використанню Java DataBase Connectivity, модулю JDBC[18] фреймворка Spring та паттерну Data Access Object (DAO).

Архітектура розподілених обчислень використовується при необхідності розв'язання трудомістких обчислювальних завдань. Вона характерна наявністю декількох екземплярів ПЗ на різних комп'ютерах об'єднаних в мережу, які співпрацюють. Впровадження моделі розподілених даних MapReduce дозволить виконувати паралельні обчислення шляхом виділення окремих невеликих задач, котрі кожна нода може виконувати незалежно.

Отже, було вирішено об'єднати декілька підходів при проектуванні архітектури програмного продукту. В вертикальному розрізі вона матиме вигляд представлений на рисунку 3.7, де чітко виділяються архітектурні рівні.

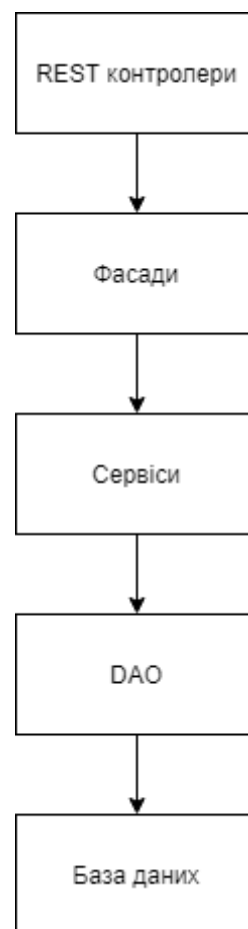


Рисунок 3.7 – Багаторівневе розподілення архітектури програми

Побудуємо діаграму розгортання. Діаграма розгортання (англ. deployment diagram) — це діаграма в UML, на якій відображаються обчислювальні вузли під

час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду.

На рисунку 3.8 представлена діаграма розгортання для системи підтримки прийняття рішень, на якій зображено, як взаємодіють її частини один з одною при виконанні дій користувачем.

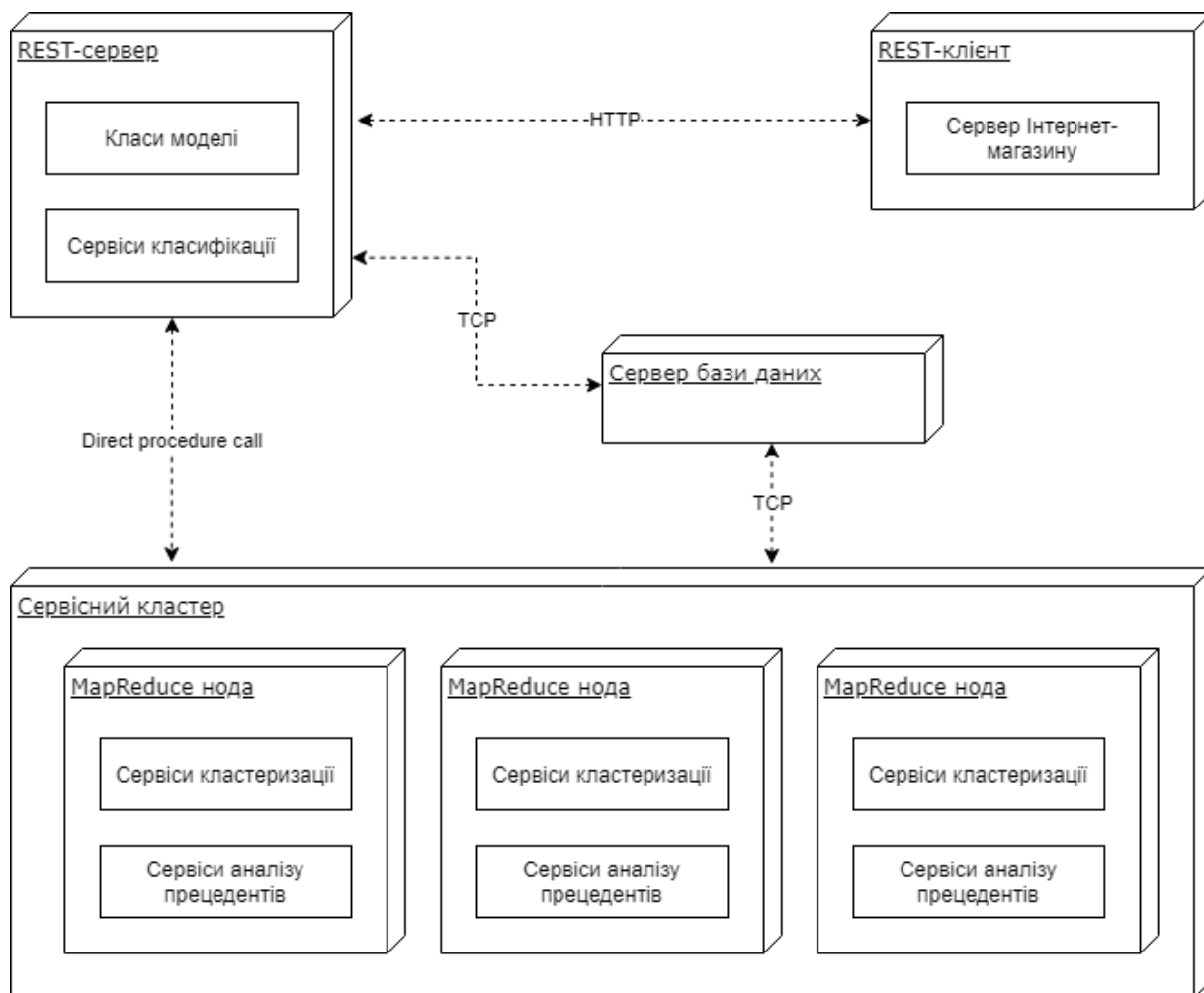


Рисунок 3.8 – Діаграма розгортання

Зокрема, на цій схемі зображено розбиття системи на окремі модульні частини – сховище даних, сервіс-ядро, кластер аналізу даних та його ноди і веб-клієнт. Представлені модулі обмінюються між собою повідомленнями використовуючи протоколи HTTP та TCP.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

4.1 Опис фізичної моделі бази даних

Реалізована система являє собою веб-додаток, який приймає набір статистичних даних, що використовується для навчання та список альтернатив на основі яких повинно прийматися рішення, котре в результаті буде відправлено користувачу. Ця система складається з Rest-сервісу для представлення інтерфейсу взаємодії з користувачем, кластеру інтелектуальної обробки даних та систему контролю базами даних.

Для збереження даних програмного продукту використовується MongoDB — документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій. У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який з реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу.

Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД простим додаванням нових машин.

Як відомо, будь-яка реляційна БД має стандартну схему, яка показує кількість таблиць і зв'язку між ними. У MongoDB такої схеми з зв'язку між таблицями немає.

Нижче наведені основні переваги MongoDB:

- відсутність схеми.
- Дана БД заснована на колекціях різних документів. Кількість полів, зміст і розмір цих документів може відрізнятися. Тобто різні сутності не повинні бути ідентичні за структурою.
- Вкрай зрозуміла структура кожного об'єкта.
- легко масштабується
- Для зберігання використовуваних в даний момент даних використовується внутрішня пам'ять, що дозволяє отримувати більш швидкий доступ.
- Дані зберігаються у вигляді JSON документів
- MongoDB підтримує динамічні запити документів (document-based query)
- Відсутність складних JOIN запитів
- Немає необхідності мапінга об'єктів додатки в об'єкти БД

Узагальнюючи можна сказати, що MongoDB є досить непоганим рішенням, якщо ми маємо справу з Big Data. Уся інформація зберігатиметься у шести колекціях: події, атрибути товарів, категорії, прецеденти та прийняті рішення.

4.2 Реалізація веб-сервісу

Веб-сервер – це частина системи, що відповідає за взаємодію з користувачем та управління процесами в роботі СППР. Для виклику методів сервера використовуються REST контролери з фреймворку Spring, що дозволяє отримувати та відправляти різноформатні дані по протоколу HTTP.

На сьогоднішній день прийнято використовувати REST – (скор. від англ. RepresentationalStateTransfer– «передача репрезентативного стану») – метод взаємодії компонентів розподіленого додатка в мережі Інтернет, при якому виклик віддаленої процедури являє собою звичайний HTTP-запит, а необхідні дані передаються як параметри запиту.

У свою чергу, HTTP – протокол передачі даних, що використовується в комп'ютерних мережах. Назва скорочена від Hyper Text Transfer Protocol, протокол передачі гіпер-текстових документів. Для ідентифікації ресурсів HTTP використовує глобальні URI. На відміну від багатьох інших протоколів, HTTP не зберігає свого стану.

Даний модуль бере на себе відповідальність за здійснення попередньої обробки даних, а саме класифікації інформації отриманої від користувача за попередньо визначеними правилами. Такий тип операції не потребує великої кількості ресурсів, а отже, не повинний являтися вузьким місцем в швидкодії роботи системи.

Для виконання ресурсоемких завдань цей модуль розподіляє наявні дані на невеликі порції, котрі можуть бути оброблені незалежно. Тобто виконує роль оркестратора, розподіляючи та делегуючи атомарні завдання.

На основі результатів роботи розподілених задач, система в повній мірі готова для здійснення своєї безпосередньої задачі, себто, прийняття рішення з урахуванням представлених альтернатив. Важливо підмітити, що кількість запитів щодо прийняття рішення може бути досить великою, та потребує можливість

обробки декількох в один момент часу, що залежить від навантаження на Інтернет-магазин.

4.3 Реалізація MapReduce ноди

Основною робочою одиницею системи є MapReduce нода. Взагалі потреба використання даної моделі розподілених обчислень пояснюється особливостями обраної предметної галузі та поставленої задачі, котра вирішуються розробкою даного програмного продукту, а саме – потенційно великим об'ємом даних, що потребує обробки. Так, об'єм даних, що використовувались при розробці системи майже сягає 1 Гб, а в реальних умовах може перевищувати декілька терабайтів. Якщо говорити про кількість записів, які належить опрацювати, то обраний датасет містить приблизно 23 мільйони входжень.

Тому система передбачає розподілення усього необхідного для аналізу об'єму даних на невеликі порції, кожна з яких може бути оброблена незалежно за невеликий проміжок часу. Першочергова задача, яка лежить перед цією програмною частиною – кластеризація представлених зареєстрованих подій за часом, тобто розрізнення сесій роботи користувача на сайті. Кожна з нод займається кластеризацією подій окремого користувача і після виконання однієї порції роботи автоматично починає роботу над іншою. Тобто ноди самостійно вибирають задачі для обробки з набору усіх наявних, таким чином вирішується проблема рівномірного розподілу завдань між потоками та їх можливого тривалого простою.

Другим типом задач, що виконують ноди є оцінка представлених критеріїв та формування інтегральних агрегатних критеріїв на основі яких створюються сутності прецедентів, котрі згодом зберігаються в завчасно підготовлену колекцію бази даних.

4.4 Інтерфейс системи

Передбачається, що система буде використовуватися сервером Інтернет-магазину, тому на заміну звичному графічному інтерфейсу приходять REST API інтерфейс. Взаємодія з СППР відбувається шляхом відправки HTTP запитів на відповідні представлені ресурси з передачею даних у форматі JSON коли це потрібно. У свою чергу, ресурси відповідають на отриманий запит певним кодом керуючись загальноприйнятим принципам. Далі розглянемо кожен з ресурсів окремо.

4.4.1 POST /data/event

HTTP REST ресурс, запит до якого здійснюється за URI `http://<домен серверу>/data/event` з вказанням методу POST. Відповідає за завантаження до системи початкових даних, а саме – списку зареєстрованих подій на сайті Інтернет-магазину. Передбачає завантаження декількох подій одночасно. Формат тіла запиту представлений на рисунку 4.1. У разі успішного завершення операції, відповідає з HTTP статус-кодом 201 Created.

```
[{
  "timestamp": <long>,
  "visitorId": <long>,
  "type": <enumeration: [view, addtocart]>,
  "productId": <long>
}]
```

Рисунок 4.1 – Формат тіла запиту POST /data/event ресурсу

4.4.2 POST /data/product-property

HTTP REST ресурс, запит до якого здійснюється за URI `http://<домен серверу>/data/product-property` з вказанням методу `POST`. Відповідає за завантаження до системи початкових даних, а саме – списку атрибутів (характеристик) продуктів. Передбачає завантаження декількох атрибутів одночасно. Формат тіла запиту представлений на рисунку 4.2. У разі успішного завершення операції, відповідає з HTTP статус-кодом `201 Created`.

```
[
  {
    "timestamp": <long>,
    "productId": <long>,
    "property": <string>,
    "value": <string>
  }
]
```

Рисунок 4.2 – Формат тіла запиту `POST /data/product-property` ресурсу

4.4.3 POST /data/category

HTTP REST ресурс, запит до якого здійснюється за URI `http://<домен серверу>/data/category` з вказанням методу `POST`. Відповідає за завантаження до системи початкових даних, а саме – інформації про категорії. Передбачає завантаження декількох категорій одночасно. Формат тіла запиту представлений на рисунку 4.3. У разі успішного завершення операції, відповідає з HTTP статус-кодом `201 Created`.

```
[
  {
    "categoryId": <long>,
    "parentCategoryId": <long>
  }
]
```

Рисунок 4.3 – Формат тіла запиту POST /data/category ресурсу

4.4.4 POST /analyze

HTTP REST ресурс, запит до якого здійснюється за URI `http://<домен серверу>/analyze` з вказанням методу POST. Відповідає за проведення аналізу попередньо завантажених даних та збереження до бази оброблених прецедентів. Даний ресурс потрібно викликати кожен раз після завершення завантаження інформації, до запиту щодо, власне, прийняття рішень. Тіло запита не передбачається і будь-яка відправлена інформація буде ігноруватися. У зв'язку з тим що аналіз даних займає багато часу, а відповідь на запит має бути надіслена якомога швидше, обробка ресурсу виконується асинхронно, а HTTP статус-код відповіді вказується 202 Accepted. У випадку, коли система вже знаходиться в процесі аналізу даних, буде відправлений код помилки 423 Locked.

4.4.5 POST /recommend

HTTP REST ресурс, запит до якого здійснюється за URI `http://<домен серверу>/data/category` з вказанням методу POST. Представлений ресурс є ідемпотентним, це значить, що система повертатиме однакову відповідь на однакові запити і її стан не змінюватиметься в результаті їх виконання. Це

найбільш цікавий для користувача ресурс, адже він безпосередньо відповідає за прийняття рішень, себто формує список рекомендацій для покупки на основі вже переглянутих товарів. В тілі запиту необхідно вказати перелік переглянутих продуктів у виді набору їх атрибутів. Формат тіла запиту представлений на рисунку 4.4. У разі успішного завершення операції, відповідає з HTTP статус-кодом 200 ОК та повертає список ідентифікаторів рекомендованих продуктів. У випадку, коли система не готова для прийняття рішень у зв'язку з незавершеним процесом аналізу даних, буде відправлений код помилки 423 Locked.

```
[
  {
    "property": <string>,
    "value": <string>
  }
]
```

Рисунок 4.4 – Формат тіла запиту POST /recommend ресурсу

5 АНАЛІЗ ДОСЛІДНИЦЬКОЇ ЕКСПЛУАТАЦІЇ ТА МОЖЛИВИХ ЗАСТОСУВАНЬ

5.1 Аналіз можливих застосувань

В ході виконання магістерської атестаційної роботи було спроектовано та розроблено програмну систему, яка неодмінно допоможе користувачам Інтернет-магазинів при здійсненні покупок. Система надає користувачу можливості отримати рекомендації при здійсненні покупок, що зумовлює підвищення шансу позитивного досвіду покупця, а це в свою чергу безпосередньо впливає на кількість придбаних товарів та збільшення прибутку для магазину. Програмний інтерфейс дозволяє серверу Інтернет-магазину легко взаємодіяти з системою, завантажувати дані для обробки, ініціювати їх аналіз та отримувати рекомендації.

В роботі було запропоновано математичну модель для області електронної комерції. Дана модель дозволяє описати особливості здійснення покупок через Інтернет-магазини, дії, які виконує покупець та рішення, які він приймає на основі певних критеріїв. Розглянута модель дає можливість розробити ефективні алгоритми для підтримки користувачів магазину при пошуку товарів шляхом представлення рекомендацій на основі вже переглянутих продуктів.

Базуючись на запропонованій моделі можна розробити алгоритми для формування пропозицій на основі характеристик розглянутих продуктів в умовах відсутності чіткого розуміння важливості кожного критерію в окремості та їх природи взагалі. В такому випадку використання прецедентів, котрі враховують раніше розглянуті продукти та рішення, що були прийняті дозволяє мінімізувати вплив суб'єктивності рішень користувачів та невизначеності критеріїв[19].

Розроблена система підтримки прийняття рішень призначення для впровадження в Інтернет-магазини середні та великих розмірів, котрі мають досить велику кількість користувачів, аби гарантувати достатність різноманіття набору даних для навчання системи. При цьому варто зазначити, що доцільність

використання розробленого продукту в магазинах аукціонного типу є сумнівною, враховуючи специфіку їх роботи.

Незважаючи на те, що від самого початку предметною областю було обрано електронну комерцію, запропонований метод прийняття рішень може використовуватися в інших сферах, де критерії вибору альтернатив мають невідому природу і приведення вектора окремих критеріїв до виду скалярного інтегрального критерію є неможливим.

5.2 Опис тестування системи

У широкому сенсі, тестування - це одна з технік контролю якості (Quality Control), яка включає планування, складання тестів, безпосередньо виконання тестування і аналіз отриманих результатів.

Важливо розуміти, що тестування ПО включає не тільки, власне, проведення тестів, але і багато інших дій, пов'язані з процесом забезпечення якості:

- аналіз і планування;
- розробку тестових сценаріїв;
- оцінку критеріїв закінчення тестування;
- написання звітів;
- рецензування документації (в тому числі і вихідного коду);
- проведення статичного аналізу.

Для перевірки якості розробки реалізованого програмного забезпечення було застосовано наступні підходи тестування:

- статичний аналіз коду;
- юніт-тестування;
- мануальне тестування API
- навантажувальне тестування

Насамперед при розробці використовувався статичний аналізатор коду SonarLint. Це безкоштовне розширення IDE, яке дозволяє виправляти помилки та вразливості під час написання коду. Як і перевірка орфографії, SonarLint висвітлює проблеми кодування під час руху з чіткими вказівками щодо виправлення, щоб ви могли їх виправити до того, як код навіть буде запущено. Приклад роботи плагіну можна побачити на рисунку 5.1.

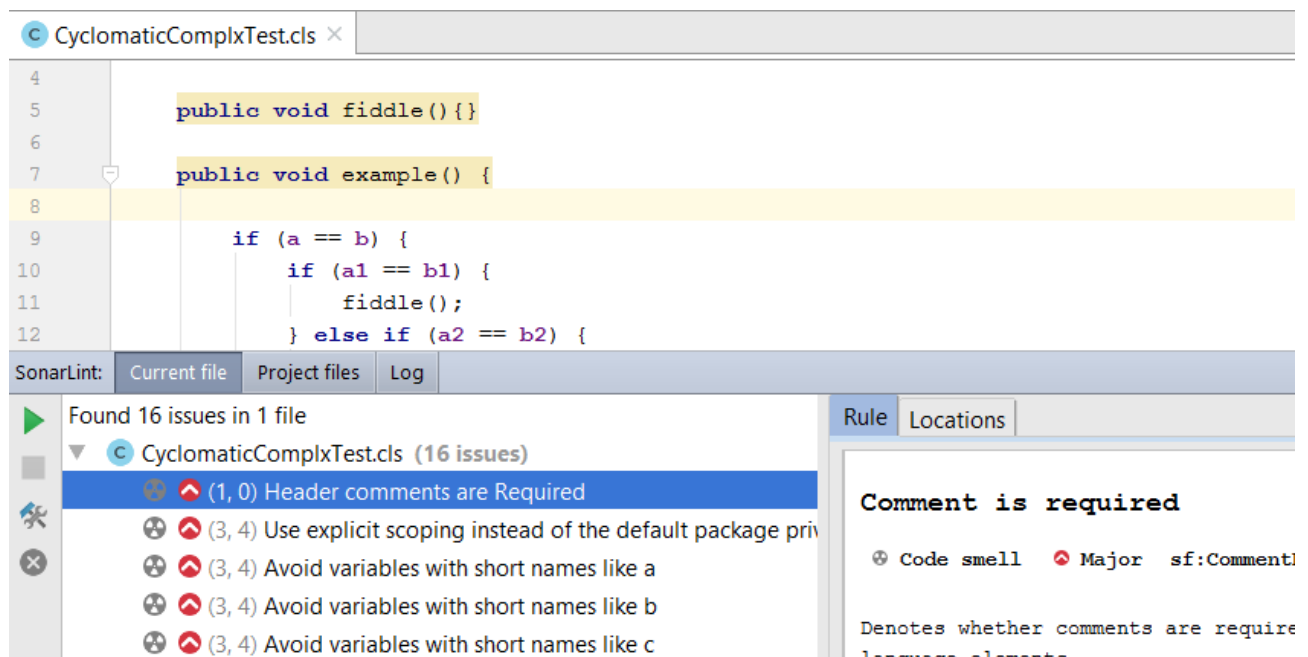


Рисунок 5.1 – Приклад статичного аналізу за допомогою Sonarlint

Модульне тестування проводилось за допомогою стандартних бібліотек мови Java – JUnit 5. Це дозволило перевірити окремі модулі програмного коду з відповідними вхідними та вихідними даними, процедурами використання та обробки.

Для перевірки функціонування системи в цілому проводилось мануальне тестування за допомогою програми Postman, яка представляє зручний інтерфейс для роботи HTTP запитамі і дозволяє писати тест-скрипти для їх тестування. Приклад використання даного застосунку представлений на рисунку 5.2. Ця

утиліта використовувалась для попереднього завантаження даних до системи, запуску їх аналізу та отримання рекомендацій на основі сформованих прецедентів.

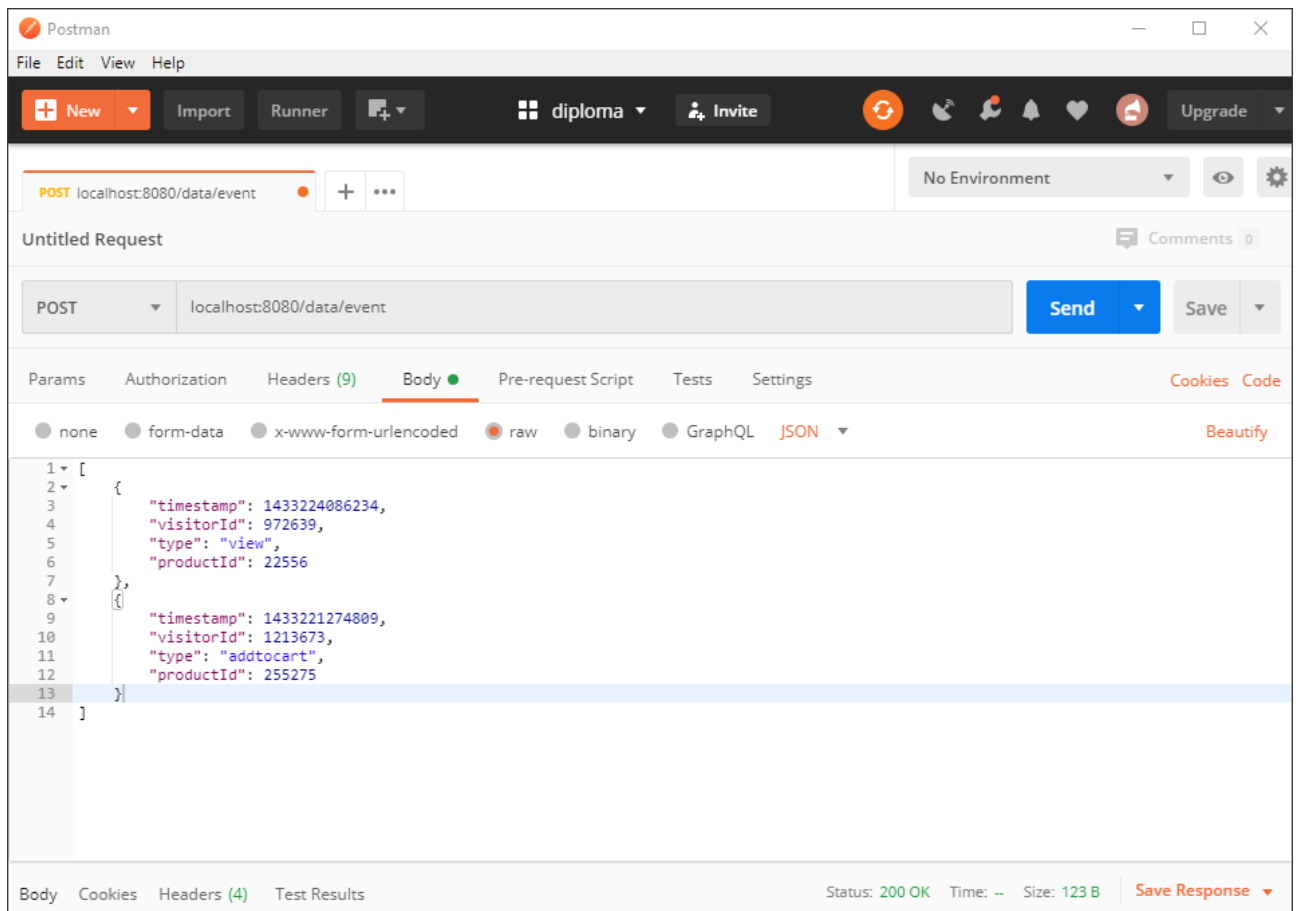


Рисунок 5.2 – Відправка HTTP запиту до системи за допомогою Postman

Одна з особливостей реалізованої системи це необхідність обробки великого об'єму даних та потреба швидкого прийняття рішень для відповіді на запити користувачів. В такому випадку велику роль грає навантажувальне тестування, яке допоможе виявити слабкі сторони системи та оцінити її можливості.

Для виконання навантажувального тестування було використано утиліту JMeter. Apache JMeter - інструмент для проведення навантажувального тестування, що розробляється Apache Software Foundation.

Хоча спочатку JMeter розроблявся як засіб тестування web-додатків, в даний час він здатний проводити навантажувальні тести для JDBC-з'єднань, FTP, LDAP, SOAP, JMS, POP3, IMAP, HTTP і TCP.

Цікава можливість створення великої кількості запитів за допомогою декількох комп'ютерів при управлінні цим процесом з одного з них. Було проведено тестування з емуляцією роботи системи за умови обслуговування Інтернет-магазином 40 користувачів одночасно. Результати тестування отримання рекомендацій (ресурс POST /recommend) зображено на рисунку 5.3.

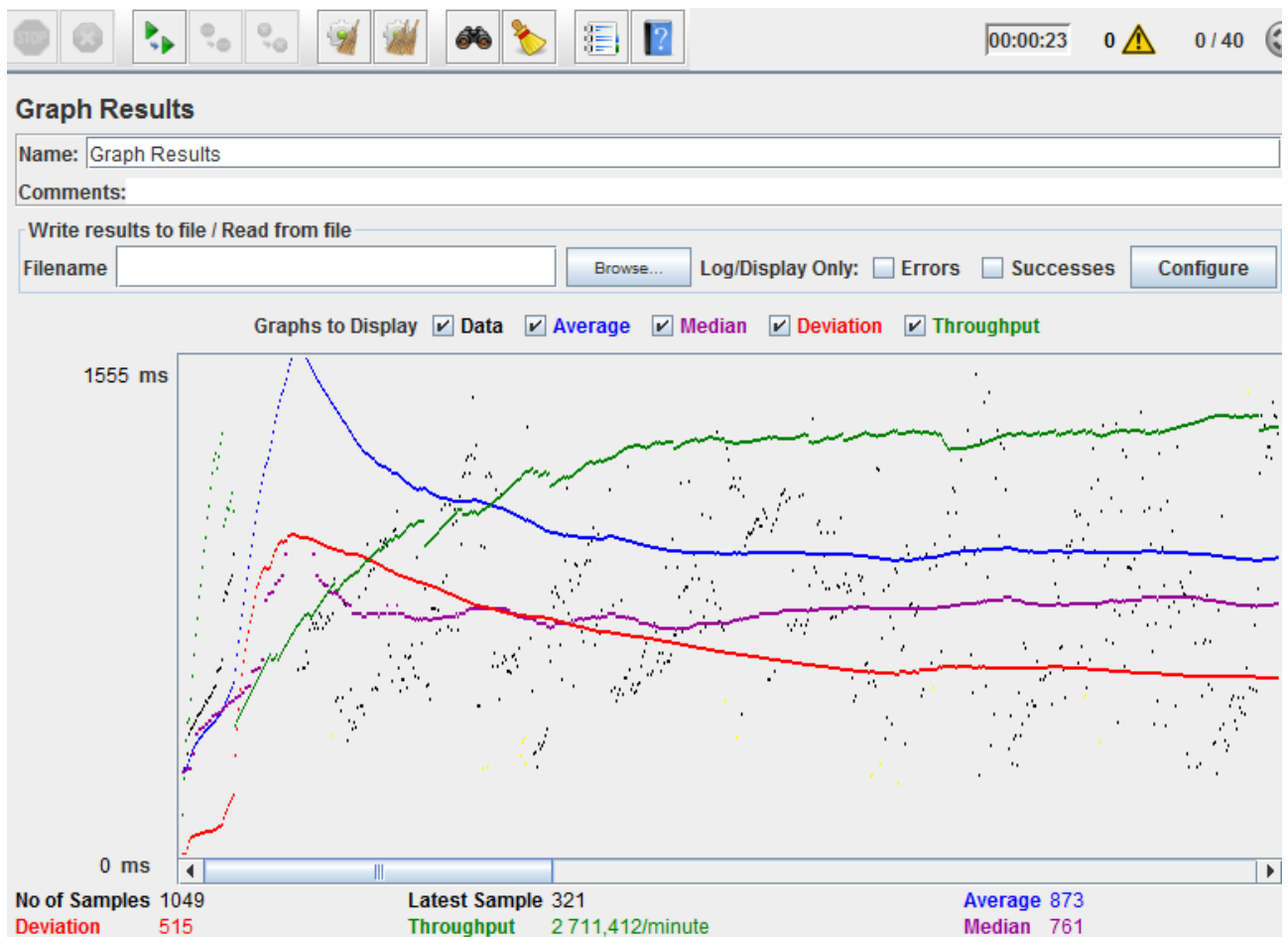


Рисунок 5.3 – Графік проведеного навантажувального тестування

Результати тестування представляють такі показники:

- Data – час відгуку на кожен окремий запит. Отримано результати між 1 с та 1,5 с;

– Average – середній час відгуку системи, об'єктивний показник навантаження.

– Median – значення медіани (корисне для використання в статистиці).

– Deviation – похибка, стандартне відхилення.

– Throughput – швидкодія виконання запиту.

Враховуючи те, що тестування проводилося на локальній машині, представлені результати є задовільними для реалізованої системи.

Таким чином, проведені тести підтвердили стабільність роботи розробленого програмного продукту та можливість його використання при багато користувальницькому доступі.

ВИСНОВКИ

В атестаційній роботі магістра було проведено аналіз методів підтримки прийняття рішень, запропоновано підхід роботи з набором критеріїв невідомої природи та реалізовано алгоритм представлення рекомендацій щодо покупки продуктів Інтернет-магазину. Був розроблений зручний інтерфейс для взаємодії з сервером магазину за допомогою HTTP протоколу. Були визначені ключові вимоги до функціоналу програмного забезпечення. Досліджено різноманітні аналоги консультаційних систем, представлені в популярних Інтернет-магазинах, вивчено їх недоліки і переваги в порівнянні з іншими.

В процесі виконання атестаційної роботи був виконаний наступний перелік завдань:

- проведено аналіз та моделювання представленої предметної галузі;
- виконано аналіз методів підтримки прийняття рішень та запропоновано відповідні модифікації, враховуючи особливості умов, в яких має працювати система;
- розроблено математичну модель для представлення рекомендацій в сфері здійснення покупок в електронній комерції
- сплановано роботу з колекціями бази даних для зберігання інформації;
- розроблено алгоритм аналізу подій та формування прецедентів і подальшого прийняття рішень на їх основі;
- виконана програмна реалізація автоматизованої СППР.

Реалізація програми проводилась за допомогою мови Java, JSON використовувався як формат передачі даних протоколом HTTP, управління залежностями для реалізація MVC підходу здійснена за допомогою фреймворку Spring. Під час роботи використовувалася IDE IntelliJ IDEA. MongoDB була обрана як єдина NoSQL СКБД.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Воронюк А., Полищук А. Актуальный интернет-маркетинг. – Київ: ІРІО, 2018. – 150 с.
2. Edwards J.S. Expert Systems in Management and Administration — Are they really different from Decision Support Systems? // European Journal of Operational Research, 1992. — Vol. 61. — pp. 114—121.
3. Сараев А. Д., Щербина О. А. Системный анализ и современные информационные технологии //Труды Крымской Академии наук. — Симферополь: СОНАТ, 2006. — С. 47-59, https://web.archive.org/web/20070928092729/http://matmodelling.pbnet.ru/Statya_Saraev_Shcherbina.pdf
4. Alter S. L. Decision support systems: current practice and continuing challenges. Reading, Mass.: Addison-Wesley Pub., 1980.
5. Перри, Б. У. Java сервлеты и JSP: сборник рецептов [Текст] / Б. У. Перри. – М.: Кудиц-пресс, 2006. - 769 с.
6. Хо К. Spring 3 для профессионалов [Текст] / К. Хо, Р. Харроп. – М.: Вильямс, 2012. – 880 с.
7. Довідковий портал по REST API [Електронний ресурс] / портал REST API Tutorial – Режим доступу: <http://www.restapitutorial.com/> –12.05.2016 р. REST API Tutorial.
8. Хемраджани, А. Гибкая разработка приложений на Java с помощью Spring, Hibernate и Eclipse / А. Хемраджани. – М.:Вильямс, 2008. – 319 с.
9. Kuzochkina A., Shirokopetleva M., Dudar Z. Analyzing and Comparison of NoSQL DBMS // International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PIC S and T 2018;
10. Дудар З. В., Широкопетлева М. С., Махін О. В. Використання методів нечіткого пошуку у реалізації веб-системи видавництва наукових журналів // Бионика интеллекта. 2014. № 1 (82). С. 99–103

11. Bonczek R.H., Holsapple C., Whinston A.B. Foundations of Decision Support Systems.- New York: Academic Press, , 1981.
12. Davis G. Management Information Systems: Conceptual Foundations, Structure, and Development. — New York: McGraw-Hill, 1974.
13. Druzdzel M. J., Flynn R. R. Decision Support Systems. Encyclopedia of Library and Information Science. — A. Kent, Marcel Dekker, Inc., 1999.
14. Eom H., Lee S. Decision Support Systems Applications Research: A Bibliography (1971—1988) // European Journal of Operational Research, 1990. — N 46. — pp. 333—342.
15. shtovba.vk.vntu.edu.ua: Нечеткий многокритериальный анализ вариантов с применением парный сравнений. URL: <http://shtovba.vk.vntu.edu.ua/file/5cf5df22175e1cb3d452cf12651ebbc0.pdf> (дата звернення 12.03.2020 р).
16. Ларичев О. И., Петровский А. В. Системы поддержки принятия решений. Современное состояние и перспективы их развития. // Итоги науки и техники. Сер. Техническая кибернетика. — Т.21. М.: ВИНТИ, 1987, http://www.raai.org/library/papers/Larichev/Larichev_Petrovsky_1987.pdf
17. Приемы объектно-ориентированного проектирования. Паттерны проектирования [Текст] / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб.: Питер, 2007. –366 с.
18. Шилдт Г. Java: Полное руководство Java SE 7 [Текст] / Г. Шилдт. – 8-е изд. – М.: Вильямс, 2012. – 1104 с.
19. Rotshtein A.P. Modification of Saaty Method for the Construction of Fuzzy Set Membership functions // Proc. of the Intern. Conf. «Fuzzy Logic and its Applications», Zichron, Israei, 1997.