Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*
(повна назва)

Кафедра *Системотехніки*
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(освітньо-кваліфікаційний рівень)

ГЮИК. 509000.005 ПЗ
(позначення документа)

«Розробка алгоритму колаборативної фільтрації для роботи з цільовою бізнес-аудиторією та його імплементація в інформаційних системах»
(тема)

Виконав: студент II курсу, групи *СПРм-18-2*
напряму підготовки (спеціальності)
122 – *Комп'ютерні науки*
(шифр і назва напряму, спеціальності)

Освітньо-професійна програма
Системне проектування
(повна назва освітньої програми)
Мамонтов Ю. В.
(прізвище, ініціали)

Керівник *проф. Ситніков Д. Е.*
(прізвище, ініціали)

Допускається до захисту

Зав. кафедри СТ _____ *проф. Гребеннік І.В.*
(підпис) (прізвище, ініціали)
2020 р.

Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*
Кафедра *Системотехніки*
Рівень вищої освіти другий (магістерський)

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

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____»_____ 20 ____ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ (ПРОЕКТ)

студентові Мамонтову Юрію Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) *Розробка алгоритму колаборативної фільтрації для роботи з цільовою бізнес-аудиторією та його імплементація в інформаційних системах*
затверджена наказом по університету від «___»_____2020 р. №_____
2. Термін подання студентом роботи (проекту) *31 травня 2020 р.*
3. Вихідні дані до роботи (проекту) Перелік використовуваних програмних засобів: ОС Microsoft Windows 10, інтегроване середовище програмування WebStorm, веб-сайт creately.com та draw.io, СУБД – MongoDB, платформа Node.js.
4. Зміст пояснювальної записки (перелік питань, що потрібно розробити) *4.1 Вступ. 4.2 Аналіз предметної області. 4.2.1 Деякі відомі рекомендаційні системи. 4.2.2 Аналітичний огляд існуючих методів. 4.2.3 Вимір точності рекомендацій. 4.3 Постановка задачі проектування. 4.3.1 Характеристика розроблюваної системи 4.3.2 Опис функцій компонентів системи. 4.3.3 Побудова діаграми прецедентів для розроблюваних компонентів системи. 4.4 Змістовий опис та аналіз використаної інформаційної технології проектування. 4.4.1 Огляд технологій проектування. 4.4.2 Побудова контекстної діаграми для обраної бізнес-функції. 4.4.3 Побудова діаграми діяльності. 4.5 Математичний опис задачі. 4.5.1 Огляд існуючих моделей аналізу соціальних мереж. 4.5.2 Математична постановка задачі. 4.6 Розробка інформаційного забезпечення системи. 4.6.1 Обґрунтування вибору СУБД. 4.6.2 Опис розроблюваної бази даних. 4.7 Розробка програмного забезпечення.*

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

5.1 Діаграма варіантів використання (1 аркуш формату А4). 5.2 Контекстна діаграма та її декомпозиція (2 аркуші формату А4). 5.3 Діаграма діяльності (1 аркуш формату А4). 5.4 Схема бази даних (1 аркуш формату А4).

6. Консультанти розділів роботи (проекту)

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	*Отримання завдання на атестаційне проектування*	*30.03.2020*	
2	*Аналіз завдання, пошук літератури та аналогів з теми атестаційної роботи*	*31.03.-05.04.2020*	
3	*Постановка задачі*	*06.04-11.04.2020*	
4	*Побудова діаграм*	*12.04.2020*	
5	*Розробка структури БД*	*13.04-16.04.2020*	
6	*Написання програмного коду*	*17.04-29.04.2020*	
7	*Розробка програмного засобу*	*29.04-30.04.2020*	
8	*Оформлення пояснювальної записки та програмної документації*	*01.05-13.05.2020*	
9	*Представлення атестаційної роботи на рецензування*	*15.05.2020*	
10	*Оформлення пояснювальної записки та програмної документації*	*16.05.2020*	
11	*Представлення дипломної роботи*	*23,05,2020*	

Дата видачі завдання 30 березня 2020 р.

Студент _____
(підпис)

Керівник роботи _____ *проф. Ситніков Д. Е.*
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до атестаційної роботи містить: 68 с., 29 рис., 2 табл., 2 додатки, 18 джерел.

Головним об'єктом є рекомендаційні системи для електронної комерції.

Мета роботи - дослідження і розробка ефективних алгоритмів для рекомендаційних систем, що дозволяють вибирати рекомендації з прийнятним рівнем релевантності в умовах великої кількості користувачів при неповній інформації про їх переваги у веденні бізнесу.

У процесі дослідження проводився аналітичний огляд існуючих методів, розробка алгоритмів на їх основі і їх реалізація.

В результаті дослідження було виявлено, що метод коллаборативної фільтрації має перевагу у вигляді точності прогнозів рекомендацій на відміну від інших методів.

Область застосування: розроблені алгоритми можуть застосовуватися в рекомендаційних системах.

Програмні рішення реалізовані в середовищі NetBeans IDE, як клієнт-серверну систему управління базами даних (СУБД) обрано MySQL. Для створення програмного забезпечення використовувалася мова серверного програмування PHP. Для функціонування клієнтської частини необхідна операційна система (ОС) Windows 10.

РЕКОМЕНДАЦІЙНА СИСТЕМА, КОЛЛАБОРАТИВНА ФІЛЬТРАЦІЯ, ІНТЕРНЕТ-МАГАЗИН, МАРКЕТИНГ, БІЗНЕС, СИСТЕМА УПРАВЛІННЯ, ІНФОРМАЦІЙНА СИСИТЕМА.

ABSTRACT

Explanatory note of attestation work contains 68 pages, 29 pictures, 2 tables, 2 additions, 18 sources.

The main object is recommendation systems for e-commerce.

The purpose of the work - research and development of effective algorithms for recommendation systems that allow you to choose recommendations with an acceptable level of relevance in a large number of users with incomplete information about their advantages in doing business.

In the course of research the analytical review of existing methods, development of algorithms on their basis and their realization was carried out.

The study found that the method of collaborative filtering has an advantage in the form of accuracy of predictions of the recommendations in contrast to other methods.

Scope: the developed algorithms can be applied in recommendation systems.

Software solutions are implemented in the NetBeans IDE, as a client-server database management system (DBMS) selected MySQL. The PHP server programming language was used to create the software. The operating system (OS) of Windows 10 is required for the client part to function.

RECOMMENDATION SYSTEM, COLLABORATIVE FILTRATION, ONLINE STORE, MARKETING, BUSINESS, MANAGEMENT SYSTEM, INFORMATION SYSTEM.

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БД — база даних;

СУБД — система управління базами даних;

ПО — програмне забезпечення;

ЦА — цільова аудиторія

ІС — інформаційна система;

ER-модель — Entity-relationship model (модель сутність-зв’язок);

PK — Primary Key (первинний ключ);

FK — Foreign Key (зовнішній ключ);

SQL — Structured query language (мова структурних запитів);

ISC — Internet Systems Consortium (вільна ліцензія для програмного забезпечення);

UML — Unified Modeling Language (уніфікована мова моделювання);

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

JSON — JavaScript Object Notation (об'єктний запис JavaScript);

MVP — Minimum viable product (мінімально життєздатний продукт);

FB — Facebook;

API — Application Programming Interface (інтерфейс програмних застосунків).

ЗМІСТ

ВСТУП

З появою Інтернету сильно виросла кількість інформації, з якою люди щодня стикаються. Це означає, що люди повинні орієнтуватися серед надзвичайно великої кількості доступних альтернатив, коли хочуть що-небудь знайти. Наприклад, від вибору нового мобільного телефону або плеєра до пошуку кінофільму для вечірнього перегляду. Що стосується самих власників інтернет-магазинів і сервісів, то вони також зацікавлені в персональній рекламі і рекомендаціях кожному конкретному користувачеві, тому що такий підхід може істотно збільшити прибуток компаній. Як результат, останніми роками інтерес до розробки і поліпшення існуючих рекомендаційних систем значно виріс.

В наші дні рекомендаційні системи вважаються одними з напрямів, що бурхливо розвиваються, вдосконалення прикладних інформаційних технологій, що є інструментом автоматичної генерації пропозицій по послугах на основі вивчення персональних потреб клієнтів. В першу чергу, рекомендаційні системи використовуються в інтернет-комерції для того, щоб допомогти користувачам вибрати відповідні товари або послуги. Такі сервіси збирають інформацію про переваги користувачів і намагаються запропонувати їм корисні товари.

Існує безліч яскравих прикладів, що використовують цей підхід. Одними з перших рекомендаційних систем були система інтернет-магазину Amazon і система компанії Google. У 1992 р. в якості основного алгоритму для рекомендаційних систем був запропонований метод коллаборативной фільтрації. Він грунтований на використанні в рекомендаційній системі інформації про доступні треки усіх користувачів. Цей метод дозволив вирішити завдання досить ефективно, виявився дуже досконалим і нині поліпшення показника якості рекомендаційної системи на 10% оцінюється в конкурсі Netflix Prize в 1 мільйон доларів.

На даний момент існує безліч методів для формування рекомендацій, але усі вони мають свої переваги і недоліки, які будуть розглянуті пізніше. Саме тому дослідження в цій області актуальні.

Об'єкт дослідження : рекомендаційні системи для електронної комерції.

Предмет дослідження : структури даних і алгоритми вибору релевантних рекомендацій.

Мета роботи : дослідження і розробка ефективного алгоритму та методу рекомендаційних систем, що дозволяють вибирати рекомендації з прийнятним рівнем релевантності в умовах великого числа користувачів при неповній або відсутній інформації про їх переваги, а також розробка архітектури системи, що використовує такі алгоритми для бізнесу. Розробка математично коректного, масштабованого і обчислювально-ефективного алгоритму рекомендаційних систем методом колоборативної фільтрації

Основні положення магістерської дипломної роботи опубліковані в 24-му Міжнародному молодіжному форуму «Радіоелектроніка та молодь у XXI столітті». Секція 2. Інформаційні системи і технології управління проектною та операційною діяльністю підприємств та організацій – Харків: ХНУРЕ. 2020.

Атестаційна робота виконана згідно з методичними вказівками [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Рекомендаційні системи змінили способи взаємодії неживих веб-сайтів зі своїми користувачами. Замість надання статистичної інформації, коли користувачі шукають і, можливо, купують продукти, рекомендаційні системи збільшують ступінь інтерактивності для розширення послуг користувачеві можливостей. Рекомендаційні системи формують рекомендації незалежно для кожного конкретного користувача на основі його минулих покупок і пошуків, а також на основі поведінки інших користувачів. Розробка рекомендаційних систем була ініційована з досить простого нагляду - люди часто покладаються на рекомендації для вирішення звичайних повсякденних завдань. Наприклад, при виборі книги для читання покладаються на поради однолітків; роботодавці враховують рекомендаційні листи; при виборі фільму люди часто покладаються на огляди і думки кінокритиків і т.д.

Рекомендаційні системи призначені для пошуку об'єктів, які сподобаються користувачу або будуть йому корисні. У типових системах є список користувачів $U = (u_1,u_2,..,u_m)$ і предметів $I = (i_1,i_2,..,i_n)$. В ході взаємодії з системою користувачі знайомляться з об'єктами, формуючи матрицю рейтингів R, де r_{uk} - рейтинг предмета $i \in I$ у користувача $u_k \in U$. Як правило, матриця рейтингів неповна і розріджена, тому що кількість різних предметів в системі велике і вже відомі u_k предмети становлять лише малу частку від загального числа. Завдання рекомендаційної системи зазвичай формулюється як обчислення передбачення і рекомендації. Користувач, для якого вони обчислюються, називають активним або міченим [2].

Передбачення - чисельне значення P_{uki}, що виражає передбачену перевагу предмета i не належить I_{uk} для активного користувача u_k. Передбачене значення лежить всередині заздалегідь визначеного інтервалу рейтингу, наприклад від 1 до 5, або заданого рівня релевантності від 0 до 1. Рекомендація - список з N деяких предметів I_r, найбільш бажаних для активного користувача, причому в нього входять лише незнайомі користувачеві елементи. Завдання в такій постановці також називають *Top-N* рекомендацією.

Перші алгоритми рекомендаційних систем використовували

рекомендації спільноти користувачів для рекомендації конкретного користувача. Рекомендувалися ті предмети, які подобалися схожим користувачам. Існують дві основні стратегії створення рекомендаційних систем: фільтрація вмісту і коллаборативна фільтрація.

При фільтрації вмісту створюються профілі користувачів і об'єктів. Профілі користувачів можуть включати демографічну інформацію або відповіді на певний набір питань. Профілі об'єктів можуть включати назви жанрів, імена акторів, імена виконавців і т.п. - в залежності від типу об'єкта. Цей підхід використовується в проекті Мусику Genome Project: музичний аналітик оцінює кожну композицію по сотням різних музичних характеристик, які можуть використовуватися для виявлення музичних уподобань користувача. При коллаборативної фільтрації використовується інформація про поведінку всіх користувачів в минулому - наприклад, інформація про покупках або оцінках. В цьому випадку не має значення, з якими типами об'єктів ведеться робота, але при цьому можуть враховуватися неявні характеристики, які складно було б врахувати при створенні профілю.

Вперше термін «коллаборативна фільтрація» був використаний в 1992 р в роботі Девіда Гольдберга [2].

У цій роботі описана система фільтрації документів, що є частиною експериментальної поштової системи Tapestry, розробленої Компан Ксерокс. Проблема було вибрати з величезного потоку розсилки щодня зростаючого числа листів, що містять гігантське число документів, тільки ті, які представляли інтерес для кожного з передплатників. Шлях прямого вибору користувачем мейл листів опинявся тупиковим через неможливість обмежитися лише невеликим числом листів, оскільки необхідні документи могли опинитися в інших. Тоді був обраний інший шлях, коли всі розсилки сканувалися автоматично і здійснювалася

фільтрація потоку, специфічного для кожного користувача. Така система сканування і фільтрації може розглядатися як рекомендаційна і розроблений в її рамках алгоритм, ліг в основу комерційних рекомендаційних систем. Математична модель, яка працює в якості основної була сформульована як задача факторизації

матриці переваг на добуток матриць чинників, що визначають індивідуальні переваги.

1.1 Деякі відомі рекомендаційні системи

Зупинимося лише на найбільш відомих з них. Напевно, найістотніший вплив на розвиток коллаборативної фільтрації зробив конкурс Netflix Prize, проведений американською компанією Netflix, що займається раніше прокатом DVD, а тепер інтернет-трансляцією фільмів за замовленням. Користувачі оцінюють проглянуті фільми за 5-балыюй шкалою, і, накопивши значну кількість оцінок, Netflix свого часу розробив рекомендаційний алгоритм Cinematch, грунтований на лінійній регресії. Накопичені оцінки були розділені на тренувальне і перевірочне (приховане) множини. Алгоритм Cinematch на перевірочній множині покращував тривіальну оцінку, розраховану як середня оцінка фільму, на 10%. Поліпшення оцінювалося згідно з середньоквадратичним відхиленням від реальної оцінки. Найцікавіше, що за поліпшення алгоритму ще на 10% Netflix оголосив нагороду в 1 млн. $ [3].

Іншою відомою рекомендаційною системою є Amazon.com Recommender. Ця система працює з найбільшим з відомих наборів об'єктів - 6 млн товарів і одним з найбільшої безлічі користувачів - 30+ мільйонів. Для генерування топ-рекомендацій використовується алгоритм коллаборативної фільтрації по об'єктах. Побудова таблиці схожих товарів в цій системі проводиться оффлайн, тобто не під час запиту і схожість товарів розраховується як косинус кута між векторами їх оцінок користувачами.

Ще одна система - Music genome Project. Це музична рекомендаційна система, грунтована на коллаборативної фільтрації вмісту. Команда експертів в цьому проекті аналізує кожну композицію і оцінює більш ніж по 400 ознакам. Ці ознаки використовуються як координатний простір для представлення кожного об'єкту рекомендацій.

1.2 Аналітичний огляд існуючих методів

Для складання рекомендацій товарів для користувача існує чотири основні підходи [5]:

- метод грунтований на утриманні (content - based);
- коллаборативна фільтрація (collaborative filtering)
- метод грунтований на знаннях (knowledge - based);
- гібридний метод (hybrid).

1.2.1 Коллаборативна фільтрація

Цей тип фільтрації будує прогнози на основі моделі вже здійснених дій на сайті або певних характеристик користувача. Ця модель може бути побудована не лише на поведінці конкретного користувача, але і з урахуванням поведінки користувачів з схожими параметрами.

Коллаборативну фільтрацію можна розділити на три типи:

- на основі сусідства (neighborhood - based);
- на основі моделі (model - based);
- гібридні моделі.

Підхід на основі сусідства є історично першим методом в коллаборативної фільтрації. Також його називають анамнестичним підходом (memory - based), тому що він формує рекомендації виходячи з оцінок, залишених користувачами при перегляді товарів або користуванні послугами сервісу. Тоді для кінцевого користувача рекомендації грунтуються за рахунок обчислення міри схожості за всіма даними, отриманих в ході аналізу оцінок користувачів. Цей метод можна розділити на два підтипи:

- на основі схожості користувачів (user - based);
- на основі схожості елементів (item - based).

User - based грунтований на порівнянні схожості користувачів итернет магазину між собою. Схожість користувач є результатом обчислень, грунтованих на оцінках, які вони дають товарам або послугам на сайті.

Для того, щоб знайти рівень оцінки рекомендації товару для конкретного користувача, необхідно пройти три кроки:

- Обчислення схожості користувачів по збігу їх переваг (вага);
- Складання списку схожих користувачів по вазі (сусідство - група);
- Обчислення оцінки товару для кінцевого користувача за оцінками "сусідів".

Для першого кроку застосовується кластерний аналіз. На скільки багато схожості і відмінності між користувачами, залежить від метричної відстані між точками значень оцінки. Для обчислення сусідства користувачів існують декілька різних алгоритмів:

- евклідова відстань;
- коефіцієнт кореляції Пірсона;
- манхэттенска відстань;
- коефіцієнт Жаккара;
- відстань Чебишева.

Евклідова відстань - геометрична відстань між двома точками у багатовимірному просторі, обчислюване по теоремі Піфагора і виглядає він таким чином (1.1) :

$$r1(X1, X2) = \sqrt{\sum_{k=1}^{m}(X1_k - X2_k)^2}, \qquad (1.1)$$

де *X1(X*2) - користувач, *k* – товар або послуга, *m* - кількість товарів або варіантів послуг в наявності, *X1k (X2k)* - оцінка першого (другого) користувача *k*- го об'єкту. Так само можна застосувати квадрат евклідова відстані для того, щоб встановити широкий діапазон ваги в схожості користувачів (1.2) :

$$r2(X1, X2) = \sum_{k=1}^{m}(X1_k - X2_k)^2, \qquad (1.2)$$

де *X1(X2)* - користувач, *k* -товар, *m* - кількість товарів, *X1k (X2k)* - оцінка першого (другого) користувача *k*- го об'єкту.

Коефіцієнт кореляції Пірсону - точніший спосіб визначення лінійної залежності між двома величинами (1.3) :

$$r3(X1,X2) = \frac{\sum_{k=1}^{m}(X1_k-\overline{X1})*(X2_k-\overline{X2})}{\sqrt{\sum_{k=1}^{m}(X1_k-\overline{X1})^2*\sum_{k=1}^{m}(X2_k-\overline{X2})^2}}, \quad (1.3)$$

де *X1(X2)* - користувач, *k* - товар, *m* - кількість товарів в наявності, *X1k(X2k)* - оцінка першого (другого) користувача *k*- го об'єкту, і - вибіркові середні *X1 і X2* відповідно. *r3(X1, X2)* може мінятися від -1 до 1. Якщо значення *r2* буде близьке одиниці, то це означає, що інтереси користувачів дуже схожі і навпаки.

Манхеттенска відстань - метрика, створена Германом Мінковським. Його так само називають city - block (street block distance), інакше кажучи, відстань міських кварталів. Цей метод часто використовують із-за простоти обчислень. Він виглядає таким чином (1.4) :

$$r4(X1,X2) = \sum_{k=1}^{m}|X1_k - X2_k|, \quad (1.4)$$

де *X1 (X2)* - користувач, *k* -товар, *m* - кількість товарів в наявності, *X1k (X2k)* - оцінка першого (другого) користувача *k*- го об'єкту.

Коефіцієнт Жаккара. Для кінцевих множин, де кожен елемент не негативний, була запропонована міра схожості Полемо Жаккаром в 1901 році (1.5) :

$$r4(X1,X2) = \frac{\sum_{k=1}^{m}\min(X1_k,X2_k)}{(\sum_{k=1}^{m}(X1_k)+\sum_{k=1}^{m}(X2_k)-\sum_{k=1}^{m}\min(X1_k,X2_k))}, \quad (1.5)$$

де *X1 (X2)* - користувач, *k* - товар, *m* - кількість товарів в наявності, *X1k (X2k)* - оцінка першого (другого) користувача *k*- го об'єкту.

Відстань Чебишева - набуває максимального значення модуля різниці відповідних ознак двох об'єктів (1.6) :

$$r6(X1, X2) = max_{k=1..m}|X1_k - X2_k|, \qquad (1.6)$$

де *X1 (X2)* - користувач, *k* - товар, *m* - кількість товарів в наявності, *X1k (X2k)* - оцінка першого (другого) користувача *k*- го об'єкту.

Методи кластеризації застосовуються на другому кроці. Розрізняють ієрархічні і не ієрархічні методи. Проте, іноді перевагу віддають введенню порогу міри близькості, як найпростішому способу. Суть якого полягає в тому, що те, хто перевищують цей поріг, називаються сусідами, а інші просто не входять до цієї групи.

Ієрархічну кластеризацію можна представити як дерево зважених кластерів (Рис 1.1).

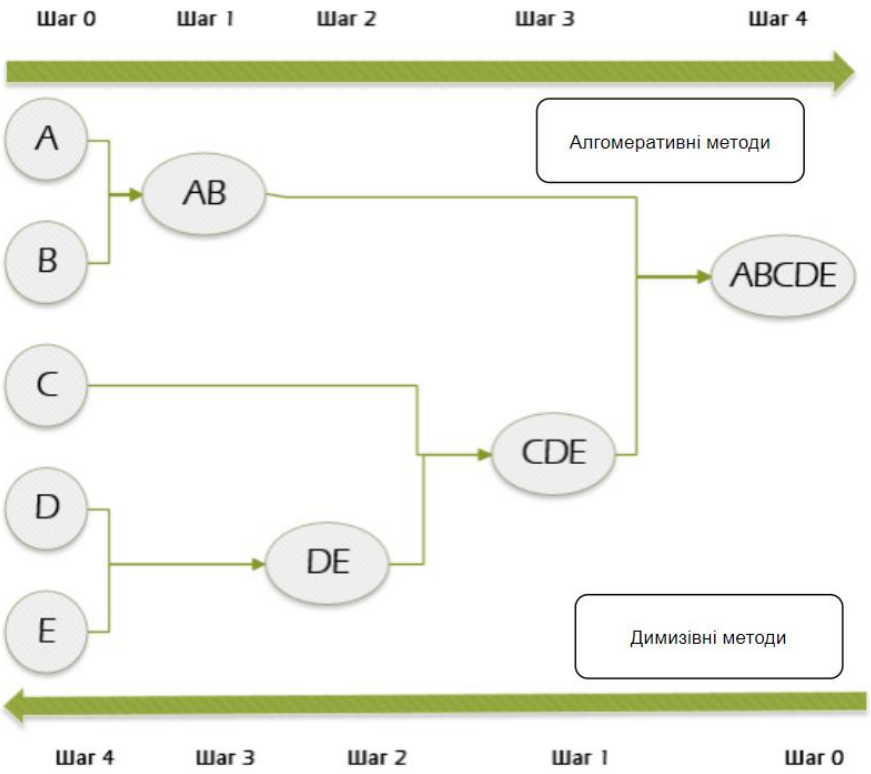


Рис. 1.1 - Приклад дерева зважених кластерів

Дерево зважених кластерів буває двох типів: агломеративним (знизу - вгору) і дивизімним (згори - вниз). У агломеративної кластеризації розбиття розпочинається з кластерів (у нашому випадку заходів схожості) тих, що містять по одному об'єкту і здійснюється послідовне об'єднання найбільш близьких кластерів. Дивизимная

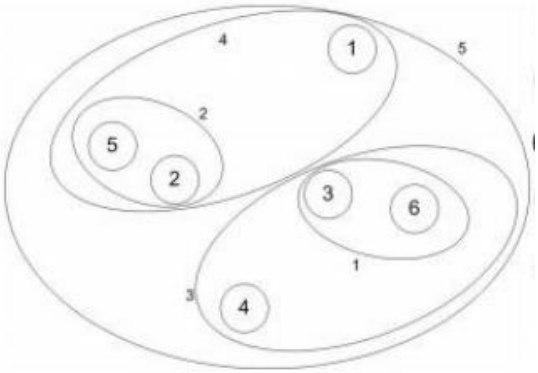
кластеризація - процедура розпочинається з одного об'єкту, де поміщені усі кластери і здійснюється послідовне відділення найбільш віддалених об'єктів. Для розрахунку відстаней між кластерами існує ряд алгоритмів :

- метод повного зв'язку (complete linkage),
- метод поодинокого зв'язку (single linkage),
- метод середнього зв'язку (average linkage),
- метод Уорду (Ward's method).

У методі повного зв'язку за відстань між кластерами приймається відстань між усіма можливими парами об'єктів, що належать різним кластерам (1.7) :

$$Kr(X1, X2) = \max r(X1, X) \quad (1.7)$$

Візуалізація кластерної структури методу (рис. 1.2 і рис. 1.3).



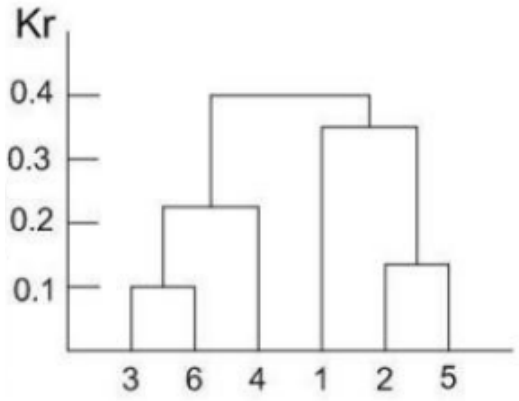


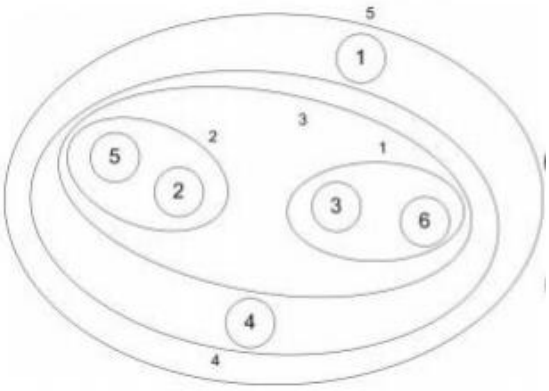
Рис. 1.2 – Вкладена діаграма

Рис. 1.3 – Дендрограмма

У методі поодинокого зв'язку за відстань між кластерами приймається мінімальне між усіма можливими парами об'єктів, що належать різним кластерам (2.8) :

$$Kr(X1, X2) = \min r(X1, X2) \quad (1.8)$$

Візуалізація кластерної структури методу (рис. 1.4 і рис. 1.5)



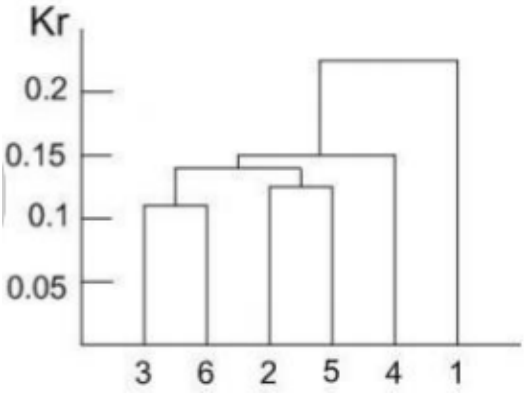


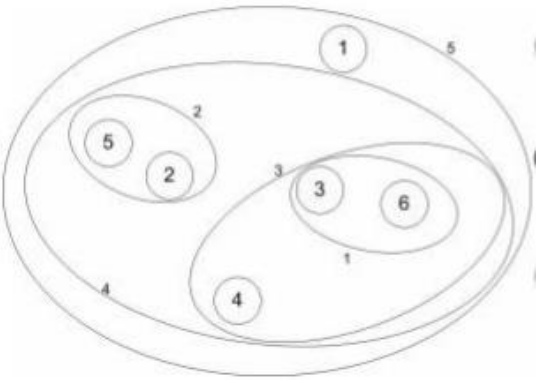
Рис. 1.4 – Вкладена діаграма

Рис. 1.5 – Дендрограмма

У методі середнього зв'язку відстань між кластерами приймається середня відстань між усіма можливими парами об'єктів, що належать різним кластерам (1.9) :

$$Kr(X1, X2) = avgr(X1, X2) \quad (1.9)$$

Візуалізація кластерної структури методу (рис. 1.6 і рис. 1.7)



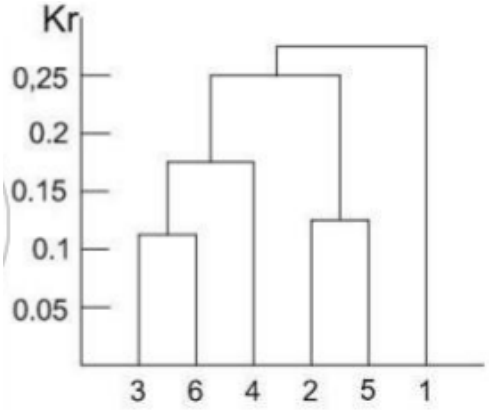


Рис. 1.6 – Вкладена діаграма

Рис. 1.7 – Дендрограмма

У методі Уорду відстань між кластерами приймається приріст суми квадратів відстаней об'єктів до центрів кластерів, що отримуються в результаті об'єднання (1.10) :

$$Kr(X1,X2) = \frac{|X1*X2|}{|X1|+|X2|} r^2 (\sum_{x1 \in X1} \frac{x1}{|X1|} * \sum_{x2 \in X2} \frac{x2}{|X2|}), \quad (1.10)$$

Візуалізація кластерної структури методу (рис. 1.8 і рис. 1.9)

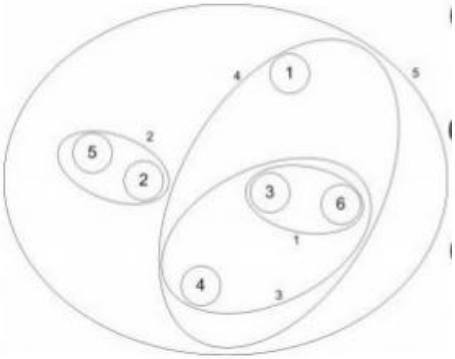


Рис. 1.8 – Вкладена діаграма

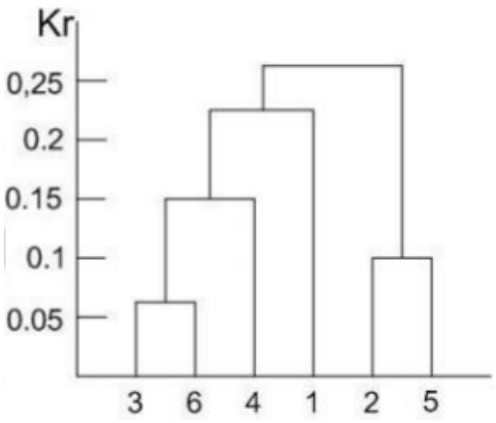


Рис. 1.9 – Дендрограмма

До не неієрархїчної кластеризацій входять такі методи, як:

- К - середніх (k - means),
- К – medoids;
- QT- алгоритм (quality threshold).

Алгоритм К - середніх:

- вибір числа До (користувачів);
- вибір початкових центрів кластерів (Х1, Х2.ХК);
- розподіл об'єктів по найближчих центрах;
- обчислення центрів кластерів;
- перевірка граничних умов;
- якщо гранична умова не досягнута, то повторювати етапи 3,4,5.

К - medoids. В якості центру кластера розглядається точка даних найбільш рівновіддалена від інших точок.

Алгоритм QT:

- вибір радіусу кластера – R,

- обчислюються кластери - кандидати. У кожному кластері-кандидатові одна з точок є центром. У кластері - кандидатові потрапляють точки що відстають від центру не більше, ніж на задану величину R;
- вибір кластера - кандидата, що містить найбільше число точок. Вибраний кластер є побудованим в точці кластера, що виключається з аналізу;
- повторити кроки 2 і 3 до тих пір, поки усі точки не будуть оброблені;

У третьому кроці повинні вичислити, яку оцінку поставить користувач, використовуючи дані "сусідів". У цьому кроці розглядаються тільки ті користувачі - сусіди, які оцінили той об'єкт, який нас цікавить. Обчислюється ця оцінка за допомогою формули (1.11) [5]:

$$O = \overline{X1} + \frac{\sum_{n=1}^{p}(X_n - \overline{X}) * r(X1,n)}{\sum_{n=1}^{p}|r(X1,n)|}, \qquad (1.11)$$

де n - один конкретний користувач з "сусідів", X_n - оцінка конкретному об'єкту n- го користувача.

Item - based грунтований на порівнянні схожості елементів. Алгоритм цього методу аналогічний до user - based, за винятком того, що тут розглядаються не користувачі, а самі об'єкти. У кінці алгоритму обчислюється середня оцінка схожих об'єктів, отримана кінцевим користувачем (1.12) :

$$O = \frac{\sum_{c=1}^{m} X_c * r(k1,kc)}{\sum_{c=1}^{m} r(k1,kc)}, \qquad (1.12)$$

де k_1 *і* k_c - вакансії. Всього m вакансій, X_c - оцінка схожих об'єктів.

Коллаборативна фільтрація на основі моделі. Ідея цього методу полягає в тому, що можна побудувати модель по сукупності оцінок, на підставі яких формуватимуться рекомендації. Модель може бути побудована за допомогою наступних методів:

- Модель Байеса;
- Методи кластерного аналізу;

- методи латентного семантичного аналізу (Latent semantic analysis - LSA);
- сингулярне розкладання (Singular value decomposition - SVD).

1.2.2 Рекомендаційна система, грунтована на утриманні

Для цього методу не потрібні дані (оцінки), які були дані користувачами об'єктам. Даний спосіб використовує інформацію про об'єкти і ретроспективну інформацію (кількість заходів в один і той же об'єкт, перегляди, скачування) про користувачів [2]. Параметри об'єктів залежать від типу. Фільтрація контенту зіставляє параметри. На основі отриманих даних він робить висновок, що користувачеві, якому подобається об'єкт А, у свою чергу який схожий на об'єкт Би, сподобається і об'єкт Б. Відповідно, для цього підходу теж застосовується заходи схожості параметрів, як і коллаборативной фільтрації.

Щоб визначити схожість об'єктів А і С можна застосувати формулу (1.13). Цю формулу так само називають коефіцієнтом Дайса.

$$rd(A_i, C_j) = \frac{|parameter(A_i) \cap parameter(C_j)|}{|parameter(A_i)| + |parameter(C_j)|}, \quad (1.13)$$

Алгоритм TF - IDF призначений для виділення ключових слів тексту. Він з'ясовує, наскільки слово має велике значення в тексті. Тут враховується частота тієї, що зустрічається слова в декількох текстах. Слово має велике значення, якщо він часто зустрічається в одному тексті і рідкісний в інших. TF (term frequency) обчислює частоту в тексті (2.14), а IDF (inverse document frequency) показує частоту документів (2.15).

$$TF(x, A) = \frac{fr(x, A)}{max_{y \in A} fr(y, A)}, \quad (1.14)$$

$$IDF(x) = \frac{N}{n(x)}, \quad (1.15)$$

де $f_r(x, A)$ - кількість слів x в документі А; N - кількість документів в наборі, $n(x)$ - кількість документів, в яких трапляється слово x.

Обчислення TF може набувати значень тільки від 0 до 1. Сам коефіцієнт TF - IDF обчислюється як множення TF на IDF. Для зіставлення двох текстів, їх можна представити у вигляді векторів у багатовимірному просторі.

1.2.3 Рекомендаційна система, грунтована на утриманні

Цей метод ідентичний з фільтрацією контенту. Він так само порівнює об'єкти. Проте аналіз об'єктів тут проходить не лише між об'єктами, але і розглядаються взаємозв'язки тих або інших груп об'єктів. Цей метод часто застосовують в онлайн - магазини. Наприклад, користувач купив ноутбук і система рекомендує йому купити до нього безпровідну мишу. Фільтрація на основі знань застосовують у тому випадку, коли інформації про поведінку користувача малі і рідкісні. Для вирішення подібної проблеми існує два методи грунтованих на знаннях:

- використання обмежень;
- вибір близьких об'єктів.

1.2.4 Гібридна фільтрація

Гібридний метод складається з двох інших фільтрацій: коллаборативна і контент. Це найскладніший і ефективніший метод з усіх методів, оскільки він закриває недолік одного методу іншим, використовуючи безліч алгоритмів.

1.3 Вимір точності рекомендацій

Для виміру точності є безліч методів, один з найпопулярніших методів - розрахунок середньоквадратичної помилки (RMSE). Після того, як на основі тестових даних алгоритм робить пророцтва, помилку можна вичислити по наступній формулі:

$$RMSE = \sqrt{\frac{1}{|T|}\sum_{(u,i)\in T}(p_{ui} - r_{ui})^2}, \qquad (1.16)$$

де u - користувач, i – об’єкт, r - оцінка, p - передбачена оцінка, T - загальна кількість тестових оцінок.

Чим менше результату, тим точніше метод складає рекомендації [6].

1.4 Недоліки та переваги

Недоліки і переваги методів рекомендаційної системи продемонстровані у вигляді схем, зображених на рисунках (1.10) і (1.11) відповідно.

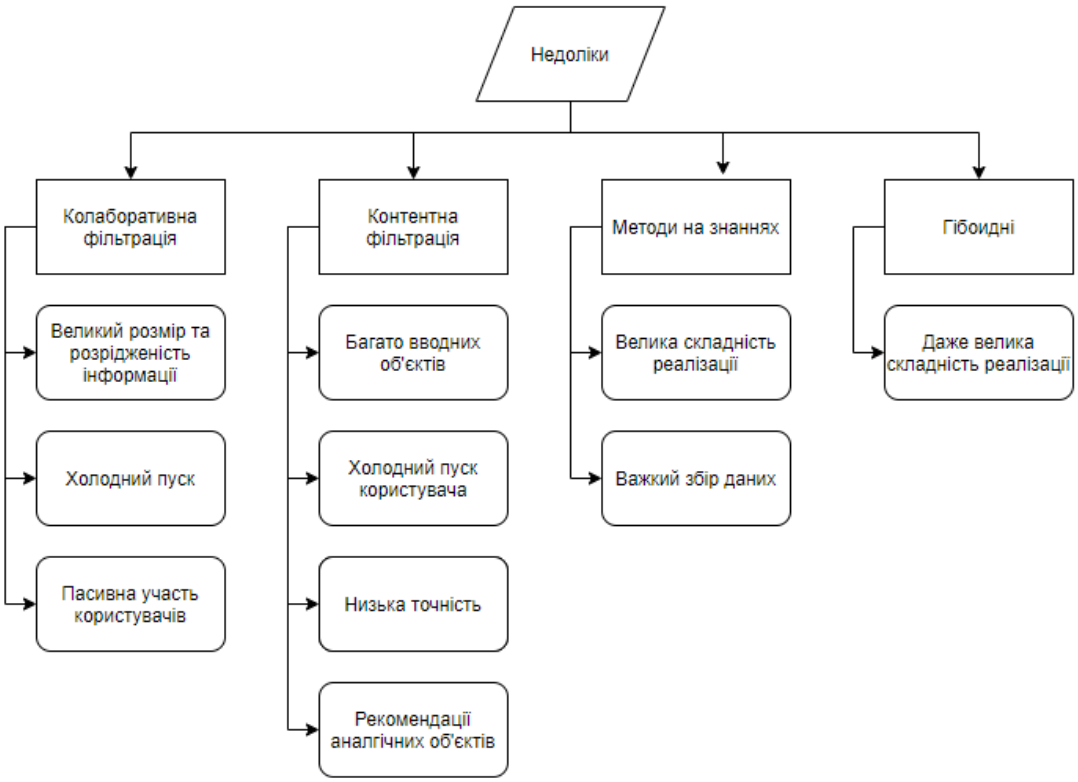


Рис. 1.10 – Недоліки методів рекомендаційної системи

Для коллаборативной фільтрації дуже багато об’єктів і користувачів і дуже мало рейтингів, оскільки користувачі оцінюють не усі об’єкти (матриця рейтингів). Система повинна вичислити інші елементи матриці. А також для нового користувача, у якого ще немає оцінених об’єктів або їх занадто мало для аналізу, складно що – або рекомендувати. Те ж саме торкається і об’єктів. Цю проблему по – іншому називають

«холодним стартом». Пасивна поведінка користувачів може привести до простою системи.

Фільтрація контенту розглядає тільки однотипні об’єкти, що добре для деяких сайтів, але іноді це невигідно і може привести до зворотного ефекту. Користувачеві можна порекомендувати однакові товари, але різних фірм. Якщо користувач купив смартфон і система почне рекомендувати інші смартфони, то це може відлякати клієнта. Ще для кожного об’єкту треба вручну вписувати дуже багато характеристик.

Також, зі збільшенням кількості користувачів в системі, з'являється проблема масштабованості. Наприклад, маючи 10 мільйонів покупців і мільйон предметів, алгоритм коллаборативної фільтрації стає занадто складений для розрахунків. Також, багато систем повинні вмить реагувати на онлайн запити від усіх користувачів, незалежно від історії їх покупок і оцінок, що вимагає ще більшої масштабованості.

Рекомендаційні системи, грунтовані на знаннях важко реалізувати, оскільки необхідно з’єднати різні групи об’єктів, щоб система не повторювала помилок фільтрації контенту. Гібридна рекомендаційна система складна тим, що в них вкладені різні алгоритми з різних методів.

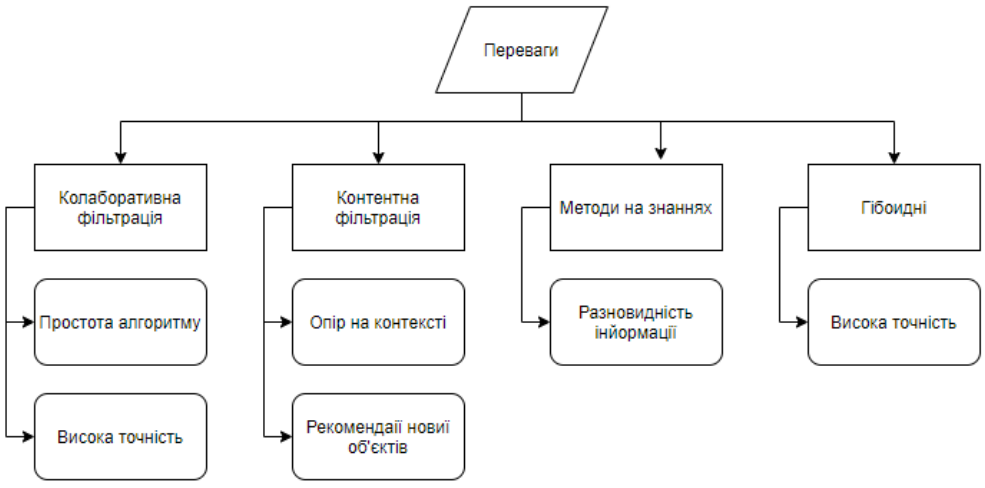


Рис 1.11 – Переваги методів рекомендаційної системи

Коллаборативна фільтрація не вимагає конкретизованих запитів. Замість цього він надає оцінки, які дають зручність користувачам, і спрощує роботу системи за рахунок простих алгоритмів. Фільтрація контенту хороша тим, що вона вирішує проблеми коллаборативної фільтрації і спирається на важливості слова в контексті документу. Рекомендаційна система, грунтована на знаннях, у свою чергу вирішує проблему фільтрації контенту.

2 ПОСТАНОВКА ЗАДАЧІ ПРОЕКТУВАННЯ

В ході написання дипломного проекту потрібно отримані наступні результати:

- Розробити алгоритм гібридної коллаборативной фільтрації, який використовує семантичну схожість інтересів користувачів, тоді як стандартні підходи використовують тільки статистичну інформацію.
- Розробити алгоритм вироблення кросс-домених рекомендацій на основі семантичного профілю користувача і коллаборативної фільтрації. Алгоритм дозволяє з необхідною точністю передбачати рейтинги, які деякий користувач присвоїть продуктам з тих доменів, в яких він раніше не виставляв рейтингів.
- Побудувати діаграми прецедентів для розроблюваної системи.

2.1 Характеристика розроблюваної системи

Предметною областю розроблюваної системи є розширення функціоналу системи і створення рекомендаційної системи, яка працює за методом колоборативної фільтрації. Розроблено з метою покращити економічні показники підприємців, зменшити час, який користувачі витрачають на пошуки та аналіз майданчиків та збільшення рівня комфорту в користування ресурсом [7]. Також кожен зарекомендований елемент повинен відповідати на ряд характеристик:

– актуальність;

– демографія ца;

– економічні характеристики ца;

– поведінка користувачів ;

– психологічні характеристики при виставленні оцінки.

Програмний засіб, що розробляється, може розглядатися як веб-сайт, який складається із компонентів (сервісів), які мають надавати можливість виконувати наступні функції:

– авторизація користувачів з допомогою електронної скриньки або соціальних мереж;

– створення, редагування завдання в особистому кабінеті;

– проведення аналізу ЦА;

– відображення основних аналітичних даних, отриманих після проведення аналізу. Вся інформація подається у списках та відображається графічно;

– біржа реклами. Відображення актуальних місць для розміщення реклами, отриманих після тематичного аналізу ЦА. Реклама у соціальних мережах, контекстна реклама;

– експорт отриманих результатів в один із популярних форматів (Excel, PDF);

– перегляд історії аналізу;

– оцінювання маркетингової платформи;

– сортування рекомендованих майданчиків для розміщення;

– отримувати аналітику.

Розроблене програмне забезпечення в основі якого лежать компоненти описані вище, буде розміщено з допомогою ліцензії відкритого програмного забезпечення «ISC». Бажаючі зможуть отримати доступ до коду та використовувати готове комплексне рішення та удосконалювати його. Доступ можна отримати на порталі GitLab [7].

2.2 Опис функцій компонентів системи

Для визначення мети розробки програмного засобу, необхідно виділити основні бізнес-функції системи. Потрібно виділити як основні функції системи, так и рекомендаційної окремо, бо все це дві складові великої системи. До них відносяться:

– проведення аналізу цільових груп користувачів за допомогою заданих ключових слів;

– розміщення та вибір «рекламних майданчиків» на біржі реклами;

– оцінювання окремих майданчиків.

Додаткові функції:

– реєстрація та авторизація користувачів через електронну скриньку соціальні мережі;

– перегляд виконаних аналізів системою;

– додавання та редагування критеріїв аналізу, які відносяться до авторизованого користувача;

– отримання інформації по завершенню, включаючи основні дані про ЦА та актуальні методи розміщення рекламних оголошень;

– отримання результатів аналізу у вигляді списків графіків та таблиць, а також їх подальший експорт у популярні формати.

– Переглядати рекомендовані майданчики;

– Сортувати та фільтрувати рекомендації;

– Додавати до «чорного списку» конкретних користувачів та майданчики;

– Ділитися з друзями успіхами в просуванні продукту або особистого бренду;

Для розроблюваної системи визначені наступні ролі користувачів:

– користувач програмного продукту — людина, яка авторизувалася використовуючи одну із соціальних мереж. Може бути майбутнім або діючим підприємцем, рекламодавцем, блогером, маркетологом, студентом або звичайним користувачем;

– модератор — людина, яка слідкує за дотриманням внутрішніх правил визначених системою.

2.3 Побудова діаграми прецедентів для розроблюваних компонентів системи

Діаграма прецедентів, також відома як діаграма варіантів використання, в уніфікованій мові моделювання відіграє важливу роль у відображенні відносин між акторами та прецедентами в системі. Основним призначення діаграми можна вважати візуально демонстрування основних функцій (варіантів використання), які допоможуть замовникам, користувачам та розробникам обговорювати систему, яка проектується або уже створена.

В першу чергу варіанти використання призначені для визначення для визначення функціональних вимог до системи та керування процесом розробки на усіх фазах розробки (ідея, дизайн, програмування, тестування, документування і так далі). Актори — це суб’єкти, які перебувають поза системою (люди, зовнішні системи або пристрої), але безпосередньо з нею взаємодіють.

Між акторами та прецедентами — основними компонентами діаграми, можуть існувати різноманітні відношення, які описують взаємодію між ними. У мові UML існує 4 основних види відношень: «залежність», «асоціація», «узагальнення» та «реалізація».

Діаграми прецедентів відіграють важливу роль при розробці програмних продуктів. Вони допомагають спростити розуміння функціональності

розроблюваного продукту за рахунок простоти подання інформації, а також надають можливість спілкуватися замовнику та розробнику «однією мовою».

Результатом визначення основних та додаткових функцій розроблюваного програмного засобу є діаграми прецедентів (варіантів використання) для кожної ролі користувача. Саме завдяки візуальній демонстрації функціональності, можна більш наглядно побачити, які функції будуть відноситися до кожної із ролей.

Для побудови діаграми прецедентів (варіантів використання) можна використовувати різноманітні засоби, у даному випадку – веб-сервіс «draw.io». Його зручний інтерфейс дозволяє створити діаграму прецедентів за лічені хвилини, експортувати файли у формат .xml або поділитися з користувачами Інтернету [8].

Для програмного засобу було розроблено дві діаграми прецедентів. На рис. 2.1 зображена діаграма для ролі «модератор», а на рис. 2.2 – для ролі «користувач».

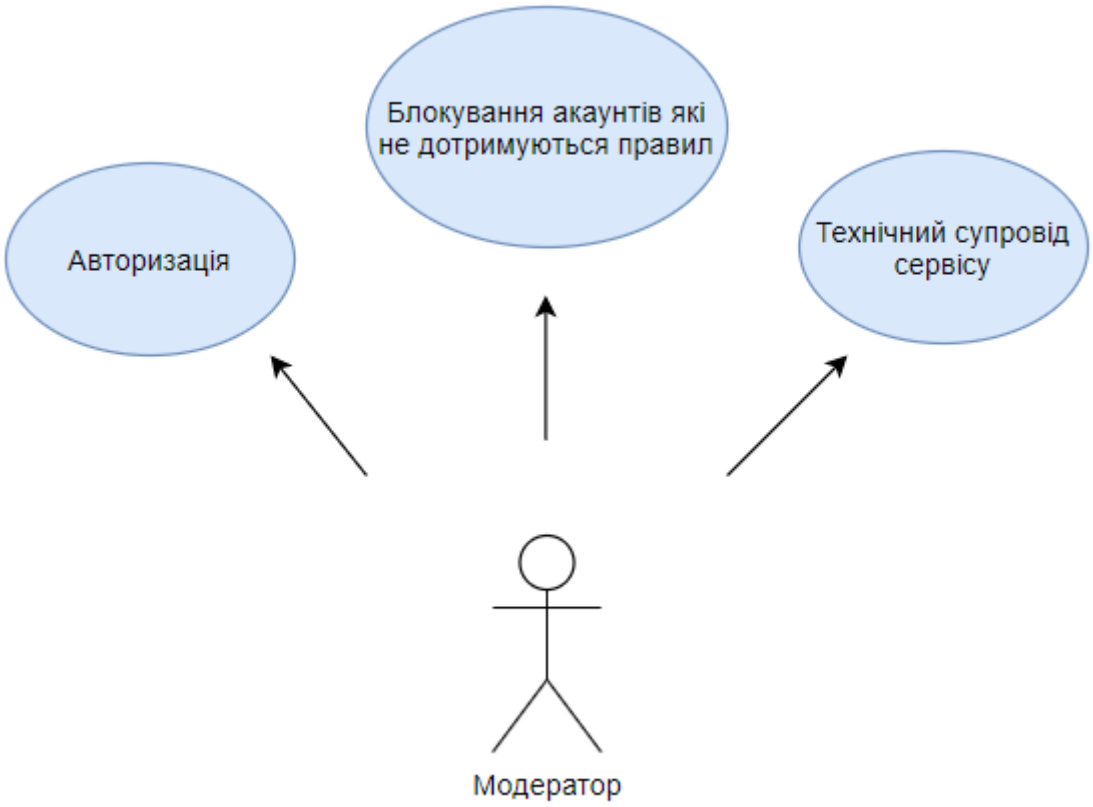


Рисунок 2.1 – Діаграма прецедентів для ролі «модератор»

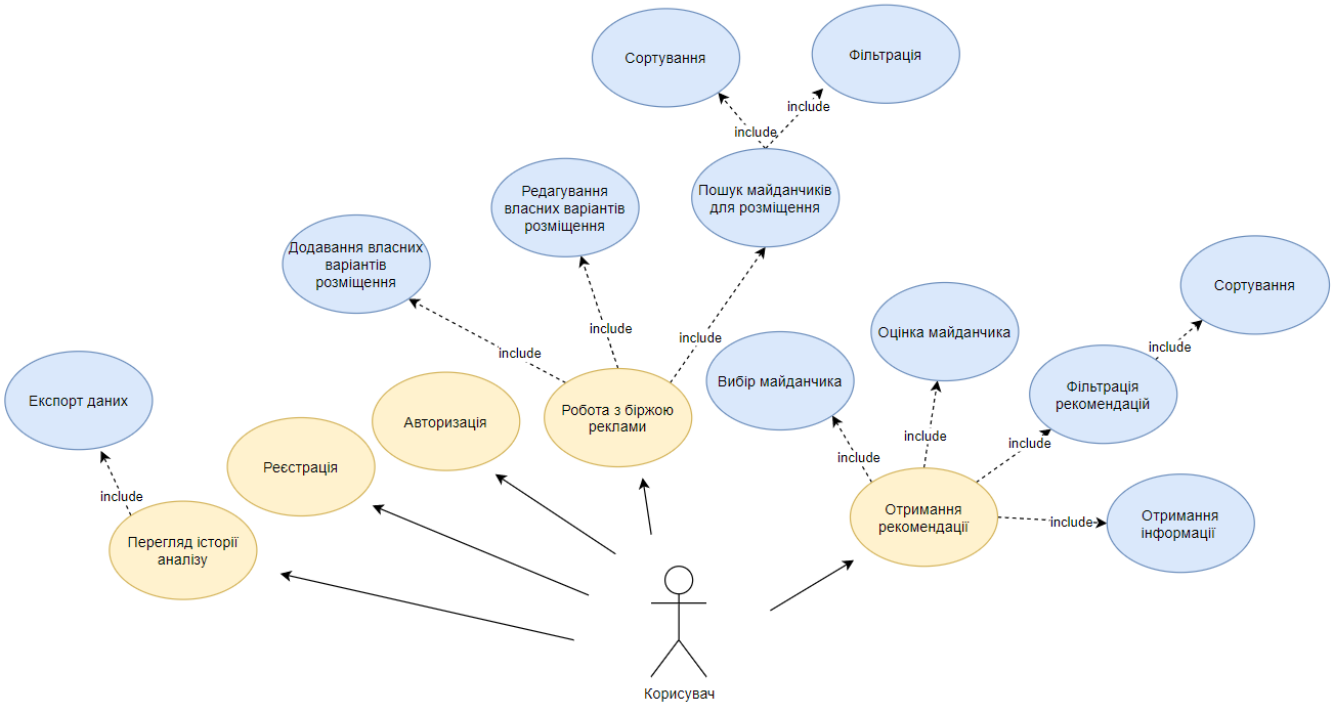


Рисунок 2.2 – Діаграма прецедентів для ролі «користувач»

На рисунках вище зображені діаграми прецедентів для розроблюваного програмного продукту. Кожна із діаграм відображає функції, які може виконувати користувач системи в залежності від прав доступу. Для ролі «користувач» визначені наступні функції:

– авторизація;

– реєстрація;

– перегляд історії аналізу ЦА, яка включає в себе:

1) експорт даних;

– робота з біржою реклами, що включає в себе:

1) додавання власних варіантів розміщення;

2) редагування власних варіантів розміщення;

3) пошук майданчиків для розміщення, що в свою чергу включає в себе:

a) сортування;

b) фільтрація.

– Отримання рекомендацій, що включає в себе;

1) Отримання інформації;

2) Вибір майданчика;

3) Оцінка майданчика після роботи з ним;

4) Фільтрація рекомендацій, що в свою чергу включає в себе:

а) сортування;

Також у системі передбачена роль «модератор», основна задача якого – слідкувати за дотриманням правил розміщення оголошень та виставлення оцінок, якщо буде спам-атака, модератор повинен швидко її зупинити. Якщо правило порушене, то місце запропоноване користувачем в якості дошки оголошення видаляється. Також модератор слідкує за дотриманням правил користування сервісом і блокує акаунти, які не дотримуються цих правил. Основними функціями модератора є:

– авторизація;

– повний технічний супровід сервісу;

– блокування акаунтів які не дотримуються правил сервісу.

Як тільки біла сформована діаграма прецедентів, розробнику та замовнику набагато зручніше оперувати набором функцій, які повинні надаватися користувачам залежно від ролі. Це ще раз показує, що створення діаграми варіантів використання є важливою частиною при розробці програмного продукту.

2.3.1 Розгорнутий опис прецеденту «Проведення аналізу цільових груп користувачів»

Для отримання чіткого уявлення про акторів, які приймають участь у прецеденті, а також про сам варіант використання потрібно описати абстракції. Опис абстракцій можна виконати у веб-сервісі «draw.io». На рис. 2.3 показаний опис абстракції «Користувач», який відображає основну суть абстракції.

Рисунок 2.3 – Опис абстракції «Користувач»

На рисунку 2.4 зображено опис абстракції «Отримання рекомендацій». Ця абстракція виконує основну задачу розроблюваної системи. Вона також включає в себе додаткові функції, опис яких зображено на рис. 2.5 – 2.7.

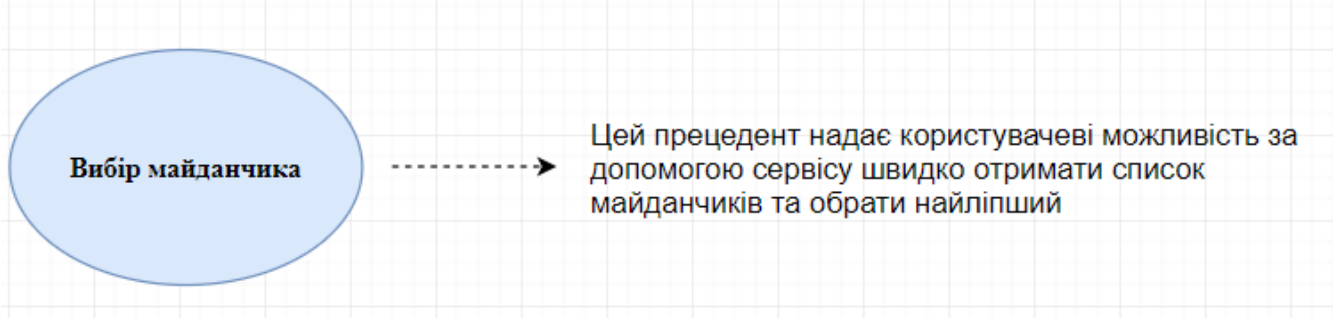


Рисунок 2.4 – Опис абстракції «Вибір майданчика»

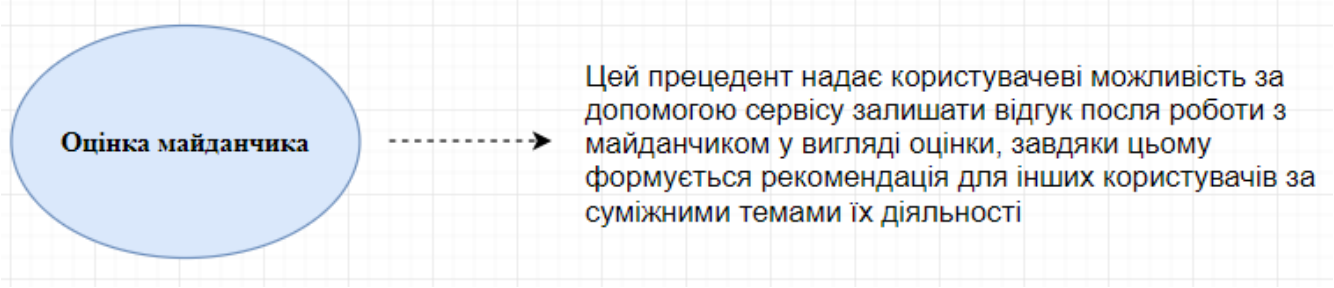


Рисунок 2.5 – Опис абстракції «Оцінка майданчика»

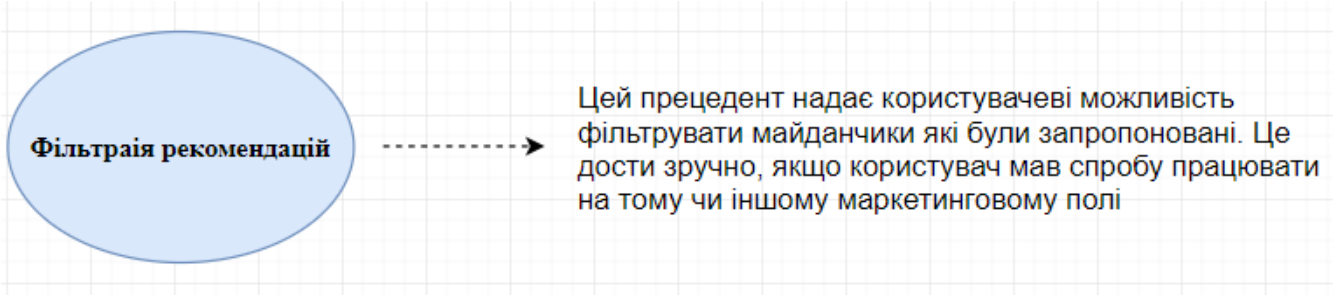


Рисунок 2.6 – Опис абстракції «Фільтрація рекомендацій»

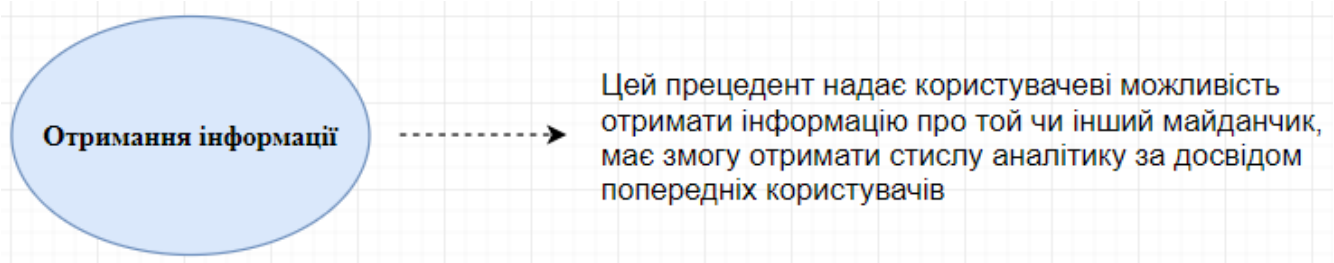


Рисунок 2.7 – Опис абстракції «Отримання інформації»

Після опису усіх абстракцій, можна виконати детальний опис обраного прецеденту.

Головний прецедент системи аналізу. «Отримання рекомендації».

Виконавчі особи: користувач.

Особи зацікавлені в експлуатації сервісу та їх інтереси:

– користувач — майбутній підприємець, зацікавлена особа. Після проходження аналізу цільової аудиторії, та отримання списку майданчиків хоче знати актуальну інформацію щодо маркетингових площадок. В цьому йому допоможе досвід колег, які мали змогу оцінити той чи інший майданчик. Користувач переслідую економію грошей та часу.

– користувач — діючий підприємець. Хоче покращити знання щодо аудиторії сайту. Це дозволить зекономити на неправильному виборі рекламних майданчиків на основі досвіду колег;

– користувач — маркетолог. Хоче покращити свої знання. Вивчити поведінку аудиторії в мережі на рекомендованих майданчиках, має змогу економити час на пошук та аналіз варіантів розміщення.

Передумови: користувач успішно авторизувався у системі з допомогою соціальної мережі.

Результат: Дані аналізу ЦА та списки рекламних майданчиків збережено до бази даних, на головній сторінці користувач отримує списки рекомендацій..

У системи є багато успішних сценаріїв. Основний успішний сценарій:

1) користувач дає оцінку майданчику та отримує схожі майданчик, які підійдуть під діяльність користувача;

а) обирає зі списку своїх рекомендацій;

б) обирає зі списку у пошуку;

2) користувач обирає майданчик після аналізу, система автоматична запропонує схожі;

3) система проводить аналіз ЦА та на основі цих даних підбирає актуальні місця для розміщення рекламних оголошень;

4) система записує отриманні дані до бази даних;

5) система на основі вибору функцій користувачем показує найактуальніші дані цільових груп користувачів або місця проведення рекламних компаній, які чітко підібрані системою під ЦА;

6) користувач на основі отриманих результатів, може виконати наступні функції:

а) подивитися стислу аналітику;

б) переглянути детальний опис груп користувачів;

в) повернутися на сторінку пошуку або на головну сторінку;

г) вийти з системи.

Альтернативні потоки. Розширений функціонал.

1) Якщо відбулась втрата з’єднання з Інтернетом.

Для того, щоб надавати користувачеві якісний сервіс, системі потрібно мати можливість відновити свою роботу з будь-якого кроку:

а) користувач оновлює сторінку;

б) система відновлює попередній стан.

2) Якщо в системі немає що порекомендувати:

а) система повідомляє про це користувача.

3) При збереженні до БД виникли проблеми технічного плану:

а) система повідомляє користувача про помилку та повторює транзакцію.

Частота використання: постійно.

3 ЗМІСТОВИЙ ОПИС ТА АНАЛІЗ ВИКОРИСТАНОЇ ІНФОРМАЦІЇ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ

3.1 Огляд технологій проектування

Технологія проектування — це сукупність технічних операцій проектування у певній послідовності і взаємозв’язку. Сьогодні на етапі проектування ПЗ можна виділити дві найбільш розповсюджені підходи розробки інформаційних систем: структурний підхід, його ще називають функціонально-модульний, та об’єктно-орієнтований. Вибір того чи іншого підходу (парадигми) має на увазі дотримання його і на стадії кодування (згідно з принципом концептуальної спільності). Їх відмінність один від одного полягає у виборі способу декомпозиції системи

Структурний підхід базується на принципі алгоритмічної декомпозиції з виділенням функціональних елементів. Згідно принципу виконується розподіл функцій інформаційної системи на окремі частини (модулі) (модуль — це логічно пов’язана послідовність команд, яка оформлена у вигляді окремого алгоритму) за функціональною належністю. Слід звернути увагу, що кожен окремий модуль реалізує один із етапів загального процесу. Структурний підхід до проектування інформаційних систем вимагає чіткого дотримання послідовності у виконанні дій. Головним недоліком цього підходу вважається його односпрямованість інформаційного потоку. Тобто, внесення змін чи виправлення помилок буде можливо тільки після повторного перегляду всієї програми. Це впливає на якість продукту, призводить до збільшення тривалості та вартості розробки [9].

У структурному аналізі використовуються в основному дві групи засобів, що ілюструють функції, виконувані системою і відносини між даними. Кожній групі засобів відповідають певні види моделей (діаграм), найбільш поширеними серед яких є наступні:

– SADT (Structured Analysis and Design Technique), також відома як IDEF0. Методологія представляє собою сукупність методів, правил і процедур, призначених для побудови функціональної моделі об'єкта будь-якої предметної області. Функціональна модель SADT відображає функціональну структуру об'єкта, тобто вироблені їм дії і зв'язки між цими діями;

– DFD (Data Flow Diagram) – діаграма потоків даних;

– ERD (Entity-Relationship Diagrams) – діаграма «сутність-зв’язок».

Якщо розробкою інформаційної системи займається група, яка включає в себе зовнішніх консультантів, експертів, постачальників ІС або у ІС відсутні описи функцій і бізнес-процесів, то методологія IDEF0 інструмент який працює. Ця методологія об’єднує всіх членів групи і дозволяє бути на одній хвилі, через це не буде виникнення непорозумінь, та час який команда витрачає на початок розробки ІС буде значно зменшений, розподіливши всі бізнес-функції між собою. Створені діаграми будуть повністю сумісні між собою і разом будуть утворювати єдину модель. Для початку побудови моделі та виявлення суперечностей існує багато програмних засобів, один із популярних BPWin фірми Computer Association.

Прерогативою даного методу є:

– проведення аналізу бізнес-процесів у системі, яку досліджують;

– метод дозволяє виявити все можливі неточності при описі системи;

– детальний аналіз виконується з використанням графічних мов моделювання (IDEF0, IDEF3, DFD) [9].

Недоліками цього методу можна вважати:

– інформація представлена у ієрархічному вигляді важко сприймається;

– якщо на ранніх етапах проектування була виявлена неточність або помилка, необхідно буде повністю переглядати програму для виправлення;

– необхідність дотримуватися чітких структур.

Після розгляду структурного підходу розглянемо ще один популярний підхід, а саме об’єктно-орієнтований. В основі цього підходу лежить принцип об’єктної декомпозиції, при цьому статична структура системи описується в термінах об’єктів і зв’язків між ними, а її поводження – у термінах обміну повідомленнями.

У проектуванні в об’єктно-орієнтованому підході головним інструментом є UML. UML (Unified Modeling Language) — уніфікована мова моделювання, основна задача якої – візуалізація та документування об’єктно-орієнтованих систем. UML включає в себе систему різних діаграм (класів, компонентів, об’єктів, пакетів, прецедентів, послідовності і т.д.), за допомогою яких може бути побудоване уявлення про систему, яка проектується.

Основною областю застосування об’єктно-орієнтованого підходу є в проектуванні складних проектів, наприклад, операційних систем, засобів розробки додатків та систем, які працюють в реальному часі. Основними перевагами об’єктно-орієнтованого підходу проектування можна вважати:

– у випадку перепроектування окремних бізнес-процесів не порушується цілісність системи в цілому;

– складові компоненти систему можна використовувати повторно;

– простота внесення змін в проекти за рахунок інкапсуляції, тобто структури об’єкта приховані від користувача, а доступ до атрибутів надається через інтерфейс;

– робота між програмістами, аналітиками, проектувальниками може бути організована паралельно;

– додатки швидко адаптуються до мінливих умов.

Недоліками можна виділити складність проведення детального аналізу в об’єктно-орієнтованому підході та неповноту певних діаграм.

Функціональне моделювання виконують за допомогою самих різних інструментів, у тому числі, не призначених для моделювання.

У кожному із основних підходів до розробки інформаційних систем як у структурному (функціональному), так і в об’єктно-орієнтованому показують себе по різному, в них існують свої переваги та недоліки. Та для того, щоб отримати максимальну ефективність під час проектування систем, слід комбінувати ці підходи опираючись на їх переваги. Під час функціонального аналізу, при проектуванні компонентів системи аналізу цільових груп користувачів інформаційного порталу, будуть розроблені контекстна діаграма та діаграма діяльності для основних бізнес-функцій [10].

3.2 Побудова контекстної діаграми для обраної бізнес-функції

Згідно методології IDEF0, аналіз бізнес-функції «Формування рекомендації» необхідно розпочати з побудови контекстної діаграми (рис. 3.1). Діаграма буде включати в себе узагальнену функцію, яка на вхід приймає інформацію про послугу, список усіх рекламних майданчиків, які застосовуються для біржи реклами, та оцінки користувачів. Формування рекомендації для користувачів буде проводитися на підставі умов та правил сервісу. Механізмом для виконання функції виступає алгоритм формування рекомендації. Результатом виконання, що свідчить про успішне завершення є чітко сформована рекомендація на основі переваг груп користувачів за суміжної цільовою аудиторією, яка подається клієнту сервісу. Якщо в системі є рекламні майданчики, які у результаті обробки системою будуть підходити під діяльність конкретного користувача, буде відображений список цих

рекламних майданчиків, що також можна вважати успішним виконанням дослідженої функції, тобто користувач сайту отримує перелік рекомендацій майданчиків для просування товарів або особистого бренду.

З допомогою контекстної діаграми, зображеної на рис. 3.1, отримати чіткий опис прецедентів з якими система буде стикатися під час виконання функцій, майже неможливо. Тому для цього зазвичай розкладають основну функцію на підфункції, тобто проводять декомпозицію діаграми.

В даному випадку на першому (нульовому) рівні декомпозиції є чотири складові (підфункції), які закладені в основу досліджуваної бізнес-функції — «Проведення аналізу цільових груп користувачів». Ними виступають: «Зчитування вхідних даних», «Обчислення схожості користувачів по групах», «Складання списку схожості користувачів», «Обчислення оцінки для конкретного користувача». Дана декомпозиція зображена на рис. 3.2.

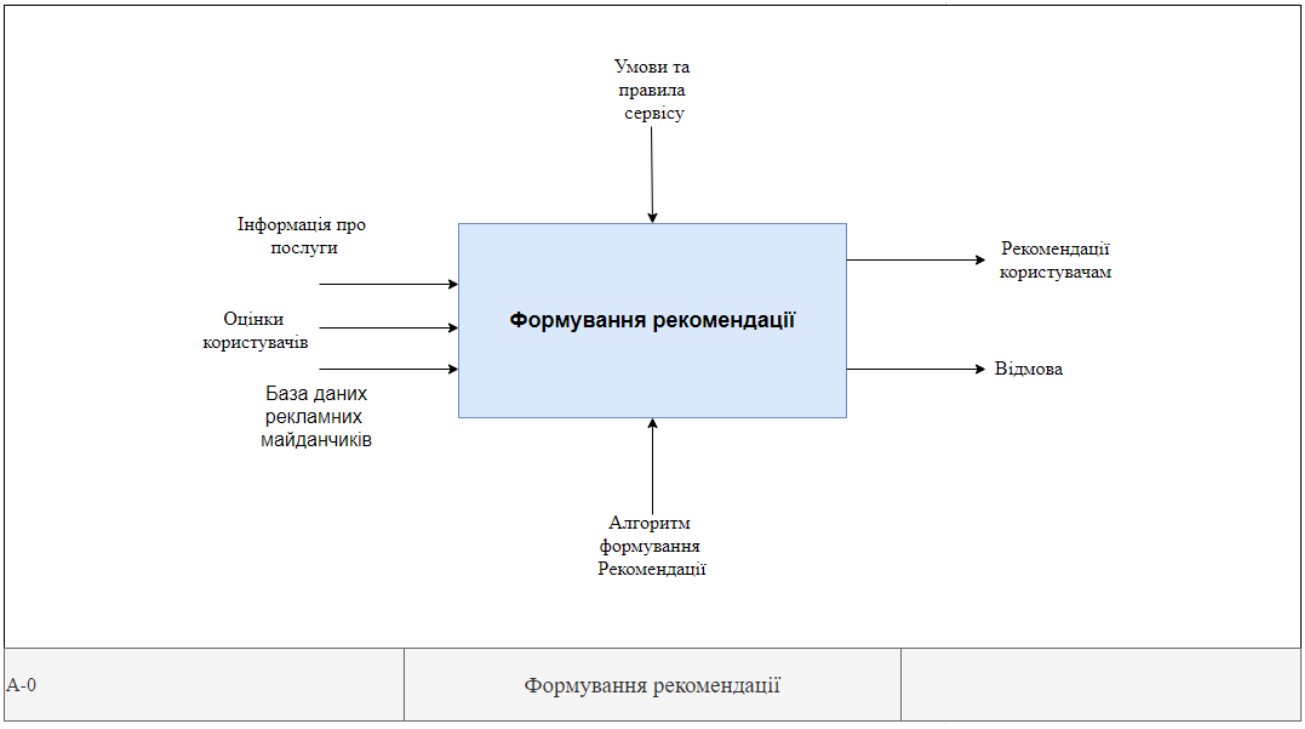


Рисунок 3.1 – Контекстна діаграма функції
«Формування рекомендації»

Початок роботи функції «Формування рекомендації», діаграма якої зображена на рис. 3.2, розпочинається зі зчитування вхідних даних. Для повноцінної роботи функції, на вхід подається інформація про послугу за яку йде мова. Якщо завдання

заповнене некоректно або не відповідає правилам проходження формуванню рекомендації, то результатом виконання функції буде відмова від проведення операції; інакше – сформована рекомендація на основі схожих користувачеві груп людей.

В результаті успішного проходження обчислення схожості, отримані на підставі правил системи дані, зберігаються у сховищі даних. Результатом роботи функції сформована група користувачів. Після того, як дані обчислення були збережені, виконується четверта функція, де обчислюється оцінка для конкретного користувача інтернет ресурсу. На вхід функції «Обчислення оцінки для конкретного користувача» подається два масиви з даними: перший — оцінка користувача, та другий — список рекламних платформ, які зареєстровані в системі. Цю функцію виконує система. В результаті на виході отримуємо чітко сформовану рекомендацію за тематикою послуги, яка сформована у вигляді списків; список рекламних майданчиків для проведення рекламних акцій, який підходить для аналізованої раніше цільової аудиторії.

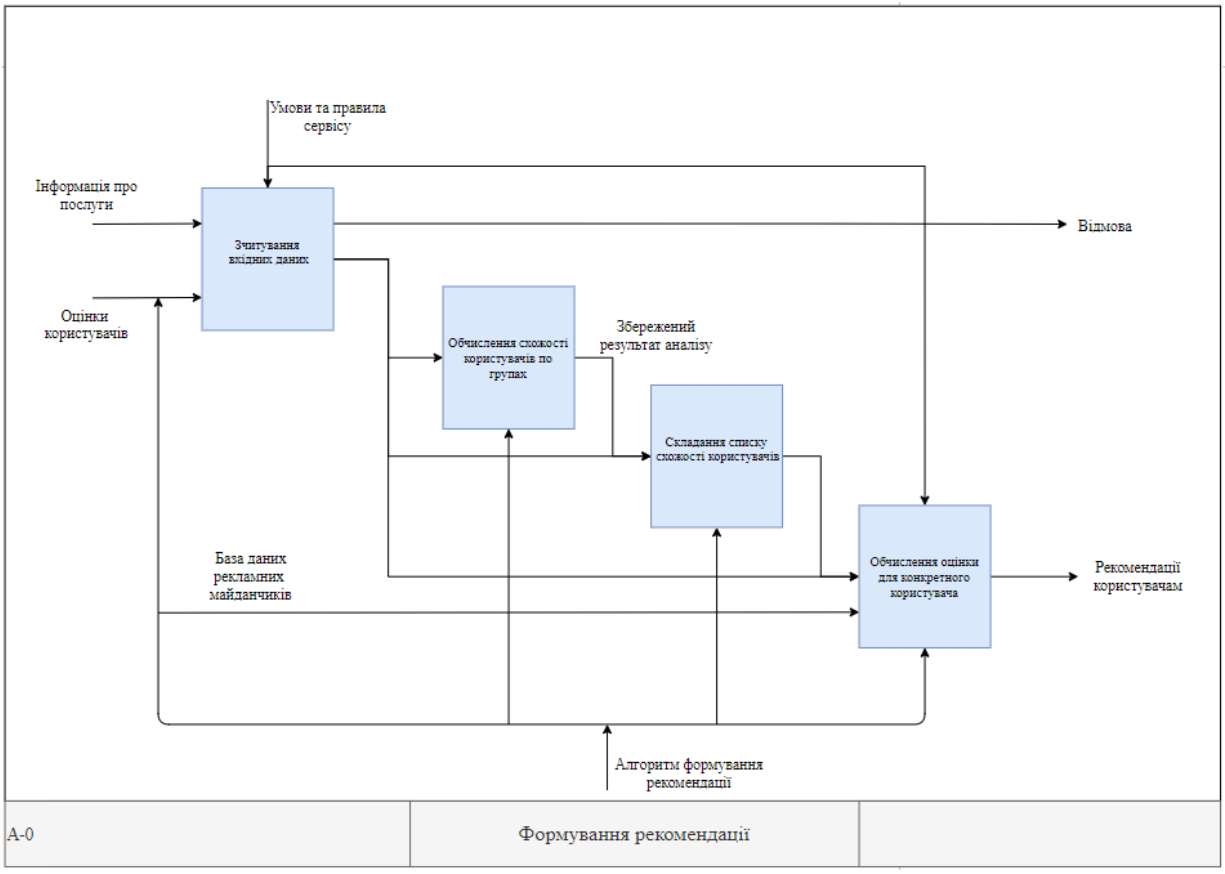


Рисунок 3.2 – Декомпозиція нульового рівня функції «Формування рекомендації»

Після проведення декомпозиції контекстної діаграми все одно неможливо до кінці зрозуміти, які процеси відбуваються у функції «Обчислення схожості користувачів по групах», «Обчислення оцінки для конкретного користувача». Для того щоб у розробника та замовника не виникало зайвих запитань, необхідно виконати ще один рівень декомпозиції, у якому буде детальне відображення процесів під час виконання функції.

Як можна побачити, алгоритм на рис 3.2 складається з чотирьох основних етапів, на виході з якого ми повинні отримати рекомендації товарів для кінцевого користувача. На вхід алгоритму подається інформація про послуги і оцінки, які зробили користувачі про ці товари та база даних рекламних майданчиків. На першому етапі важливо, який користувач поставив оцінку, якого саме товару і скільки балів становить його оцінка. Для обчислення схожості користувачів, за випадковим збігом інтересів на основі оцінок, було прийнято використовувати формулу розрахунку евклидової відстані (2.1). Цей метод розраховує вага всіх збігів для подальшого складання списку схожих між собою по інтересам користувачів.

Проведемо декомпозицію функції «Проведення процедури обчислення схожості користувачів» на шість підфункції: «Вибір числа К користувачів», «Вибір початкових центрів (X_1, X_2...)», «Розподілення об'єктів за близькими центрами», «Перерозрахунок центрів кластерів», «Перевірка граничних умов», «Сопоставлення списку груп користувачів». Декомпозиція зображена на рис. 3.3.

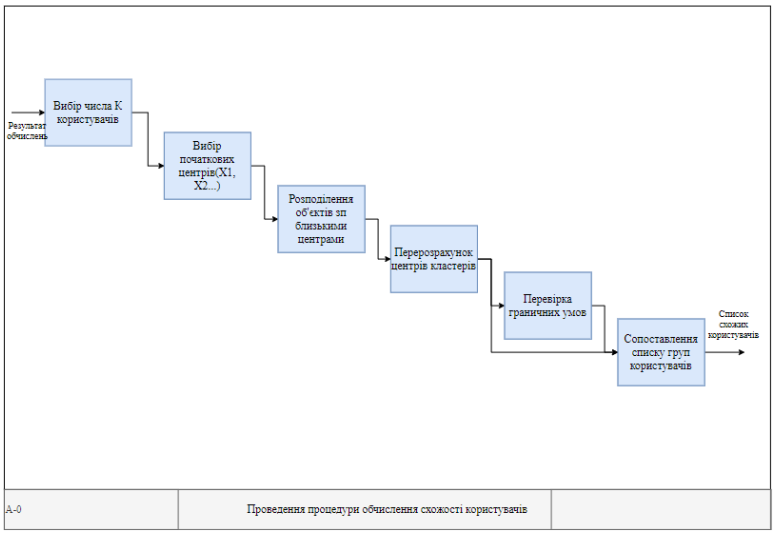


Рисунок 3.3 – Декомпозиція функції
«Проведення процедури обчислення схожості користувачів»

Після виконання декомпозиції функції «Проведення процедури обчислення схожості користувачів» можна побачити послідовність роботи методу K-средніх, який був обраний для кластеризації всіх користувачів в групи. На вхід подаються результати обчислень схожості їх інтересів, вироблених в попередньому процесі до декомпозиції. Для числа До користувачів вибираються початкові центри кластерів. Всі користувачі діляться на групи навколо цих початкових центрів в залежності від того, до якого центру ближче вони знаходяться. Далі йде переобчислення центрів кластерів для більш точного розподілу на групи. Робиться це до тих пір, поки не перестане змінятися внутрі-кластерна відстань, тобто до тих пір, поки відстань від самого віддаленого користувача до центру кластера не перестане змінятися.

Ще для більш повного розуміння проведемо декомпозицію функції «Обчислення оцінки для конкретного користувача» на три підфункції: «Отримання результатів формування груп користувачів», «Обробка даних для збереження». Декомпозиція зображена на рис. 3.4.

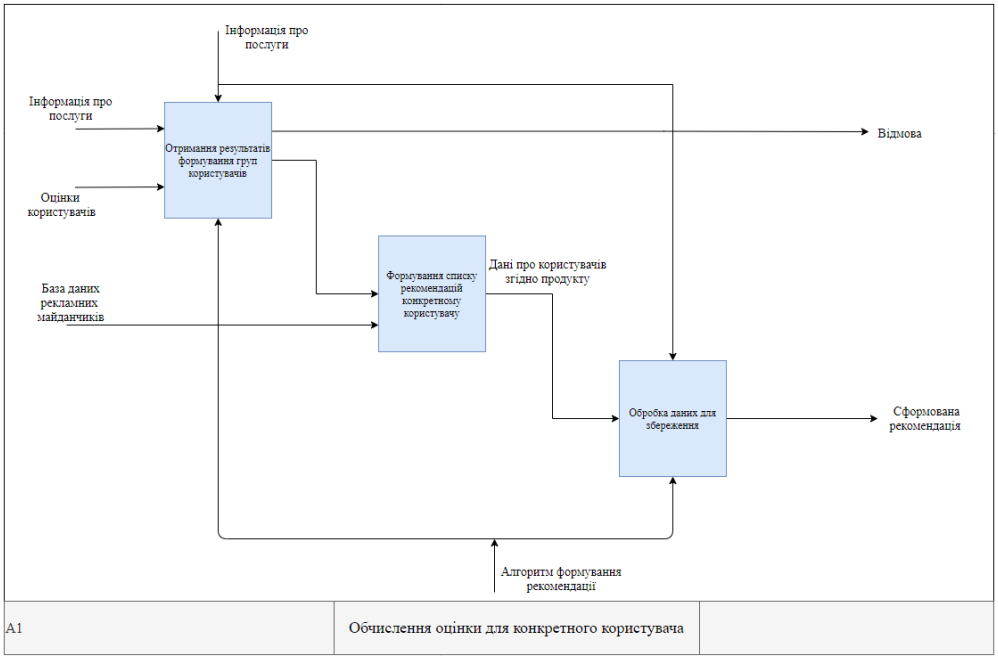


Рисунок 3.4 – Декомпозиція функції «Отримання результатів аналізу цільових груп»

Перш за все, проходить перевірка правильності та відповідності отриманих із сховищ дані. На вхід функції «Перевірка даних» подається ці самі дані. Успішним виконанням функції можна вважати масив, який система отримує із сховища даних.

Якщо під час виконання щось пішло не так і система не змогла знайти необхідні дані, то система зробить це ще раз.

Як тільки отримані дані пройшли перевірку, також на вхід подається база даних рекламних майданчиків зареєстрованих у системі, виконується наступна функція - «Формування списку рекомендацій конкретному користувачу». Успішним результатом можна вважати масив даних про користувачів згідно з продукту до якого було виявлено інтерес.

Наступний етап — це функція «Обробка даних для збереження». На вхід даної функції подаються масиви даних про користувачів, які система отримала як результат на попередній функції. Успішний результат виконання даної функції є сформований список рекомендація для користувачів сервісу.

Після проведення ряду декомпозицій бізнес функції, все встає на свої місця, і стає повністю зрозуміло про призначення кожного з функціональних блоків бізнес-функції «Формування рекомендації». На цьому етапі можна вважати, що декомпозиція успішно проведена і може бути завершена.

В результаті декомпозиції обраної бізнес-функції, було отримано повну, детальну інформацію про підфункції у вигляді ієрархічних діаграм. Це представлення дозволяє розробникам та замовникам мати повне представлення о бізнес-процесах всередині системи, та призводить до того, що між ними не виникає питань, щодо розуміння о призначенні функції та системи в цілому.

Отже, можна зробити висновок, що побудова діаграми IDEF0 для розроблюваної системи є важливим етапом у проектуванні програмного засобу [10].

3.3 Побудова діаграми діяльності

Коли йде процес розробки системи потрібно моделювати поведінку системи, і при цьому моделюванні може виникати необхідність деталізувати особливості алгоритмічної та логічної реалізації операцій. За традицією це відбувається з використанням блок-схем або структурних схем алгоритмів. Такі схеми акцентують увагу на послідовності виконання певних дій, які у сукупності призводять до отримання бажаного результату.

Для моделювання процесу виконання операцій у мові UML використовують діаграму діяльності, яка зображується графом, вершинами якого є стани (дій і/або видів діяльності), а дугами – переходи від одного стану (дій/виду діяльності) до

іншого стану (дій/виду діяльності). Діаграма діяльності для обраної бізнес-функції — «Отримання рекомендації через оцінку» зображена на рис. 3.5.

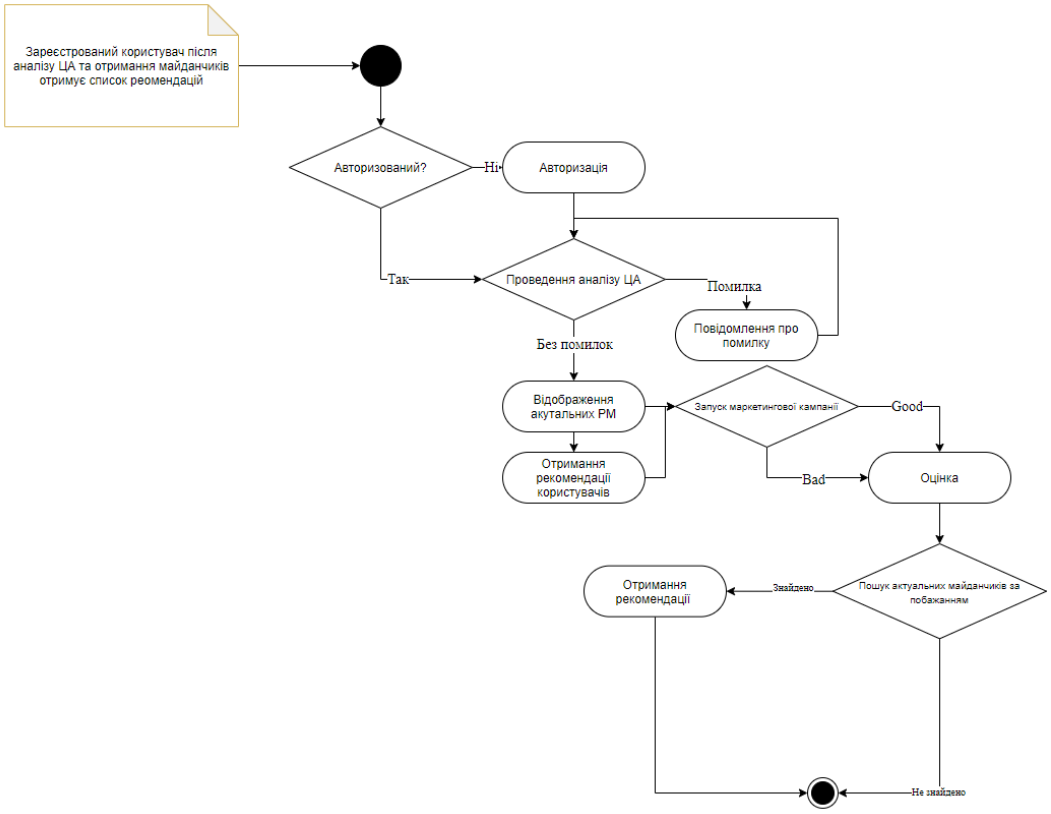


Рисунок 3.5 — Діаграма діяльності функції «Отримання рекомендації через оцінку»

4 РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕСПЕЧЕННЯ

4.1 Обґрунтування вибору СУБД

В наш час у світі існують компанії, які представляють велику кількість програмних рішень для збереження даних. Популярними серед існуючи є: MySQL, PostgreSQL, MSSQL, MongoDB [11].

Для того щоб не помилитися при розробці програмного засобу, потрібно зробити правильний вибір бази даних, а для цього необхідно розглянути кожне з представлених рішень, звернути увагу на їхні переваги та недоліки.

Postgres — об’єктно-орієнтована система управління базами даних (СУБД), також відома як PostgerSQL, походить від пакету Postgre, написаного в Berkeley. На сьогоднішній день, PostgerSQL — це одно з самих прогресивних СУБД з відкритими вихідними текстами. Використання можливо майже скрізь, СУДБ пропонує багатоваріантне керування паралелізмом, вона підтримує майже усі конструкції SQL (включаючи вкладену вибірку, транзакції та визначені користувачем типи і функції). Також особливістю цією СУБД є те, що вона має обширний зв’язок із різноманітними мовами програмування, такі як C, C++, Java, Perl, Tcl, та Python). PostgerSQL побудований на базі мови SQL.

За допомогою відкритого коду, на базі ядра PostgerSQL було розроблено велику кількість інших рішень: Amazon Redshift, HP Vertica, Netezza, Greenplum та інші. Майже кожне рішення позиціонує себе як хмарні сервіси, які орієнтовані безпосередньо на збереження та аналіз «Big Data».

Для повноцінного обзору, потрібно звернути увагу на сильні сторони даної СУБД. До них можна віднести такі особливості:

– СУБД має змогу підтримувати БД майже необмеженого розміру;

– успадкування;

– потужні та надійні механізми транзакцій та реплікацій (створення копій даних та їх синхронізація між собою, що гарантує відмовостійкість та збереження даних у разі пошкодження одного або кількох серверів із БД);

– СУБД легко масштабується, тобто моє можливість розширюватися;

Розглянемо наступного яскравого представника реляційних баз даних, і це є СУБД MySQL. Одразу подивимось на переваги. До них можна віднести наступне:

– в один момент часу, паралельно можна виконувати кілька запитів;

– записи які створюються мають фіксовану та змінну довжину;

– підключення до однієї з мов здійснюється за допомогою драйверу, який йде у комплекті;

– система привілей та паролів дуже гнучка;

– СУБД заснована на потоках і через це має швидку систему пам’яті;

– Має до 16 ключів у таблиці де кожний ключ, в свою чергу, може мати до 15 полів;

– усі поля мають значення за замовчуванням;

– операції під час роботи з рядками не звертають уваги на регістр символів, якщо порівняти з СУБД вод корпорації Oracle, то це значна перевага;

– також перевагою можна виділити простоту керування таблицею. Також сюди можна включити додавання та видалення ключів і полів.

Microsoft SQL Server (MSSQL) є однією з найбільших реляційних СУБД у світі. Історія розвитку та досвід у використанні цієї СУБД зробили це рішення одним із надійних, в деякому сенсі універсальних та швидких СУБД.

Створення критично важливих інтелектуальних програм для швидкої обробки транзакцій з обширними можливостями масштабування не стане проблемою, бо Microsoft SQL Server дозволяє це робити.

Також екосистема SQL Server дозволяє проводити обширні процеси бізнес аналітики. Провести аналітику можуть як користувачі які підключені до мережі Інтернет, так і без підключення, та навіть на мобільних пристроях.

Національний інститут стандартів та технологій США (NIST) вже 7 років поспіль визнає MSSQL найменш вразливою серед аналогів.

Основною областю використання Microsoft SQL Server залишається десктопні та веб-додатки, які розроблені з використанням технологій Microsoft.

MongoDB — наступний яскравий представник систем зберігання даних. Варто зазначити, що дана СУБД є документно-орієнтована з відкритим вихідним кодом, яка не потребує опису схем та таблиць. Ключової відмінністю від реляційних БД, які класифікуються як SQL бази даних, MongoDB — це NoSQL база даних, яка використовую JSON-подібні документи та схему бази даних. Дана СУБД написана на мові програмування C++, що дозволяє їй працювати в декілька разів швидше за інші реляційні БД [].

Переваги MongoDB:

– швидкість запису документів;

– повнотекстовий пошук;

– відмовостійкість;

– простий шардінг (розподіл великої таблиці на маленькі за певними методами, частіше за все розподіл проходить в алфавітному порядку) та реплікація;

– немає структури;

СУБД можна використовувати для веб-додатків, які написані будь-якою мовою програмування.

NoSQL рішення швидко виконують задачі запису та читання даних, тож для розроблюваного програмного засобу було прийнято рішення використовувати саме документно-орієнтовану СУБД – MongoDB [12].

4.2 Опис розроблюваної бази даних

Для розроблюваної системи аналізу цільових груп користувачів планується розробити документно-орієнтовану модель системи. Для цього необхідно створити такі сутності: User, SearchRezult, Advertisement, Task, AnalyzeRezult, AdStations, Information. Якщо поглинути в теорію, то стане відомо, що у документно-орієнтованої СУБД кожна сутність є колекцією документів. Діаграма, зображена на рис. 4.1, якраз відображає сутності та зв’язки між ними.

Для того щоб отримати інформацію про сутності програмного засобу в повному обсязі, наведемо детальний опис атрибутів сутностей та стислий опис самих сутностей (табл. 4.1):

– User — це користувач системи який пройшов авторизацію та має повний доступ до наступних функцій: в першу чергу користувач на сервісі може почати проводити власний аналіз цільової аудиторії, шукати вже завершенні результати аналізу інших користувачів. Користувач має змогу ставити оцінки після роботи в той чи іншій площадці, він може отримувати рекомендації на основі попередніх пошуків та досвіду колег;

– SearchRezult — сутність, яка зберігає в собі результати пошуку користувачів сервісу;

– Advertisement — це сутність, яка зберігає в собі усю необхідну інформацію про рекламний майданчик. Після аналізу цільової аудиторії, категорія користувачів, які по тематиці зазначеної в завданні підходять, показуються користувачеві;

– Task — сутність, яка зберігає у собі всю необхідну інформацію для проведення аналізу;

– AnalyzeRezult — це головна сутність, яка зберігає в собі дані отримані після аналізу цільової аудиторії;

– AdStations — сутність, яка зберігає в собі інформацію пор біржу реклами;

– Information — сутність, яка зберігає в собі базові відомості про інформацію цільової аудиторії після аналізу.

– AdRating — це сутність яка несе у собі інформацію про виставлений рейтинг майданчику.

– Product — сутність створена безпосередньо для виставлення рейтингу. Несе у собі головну інформацію рекомендованих майданчиків.

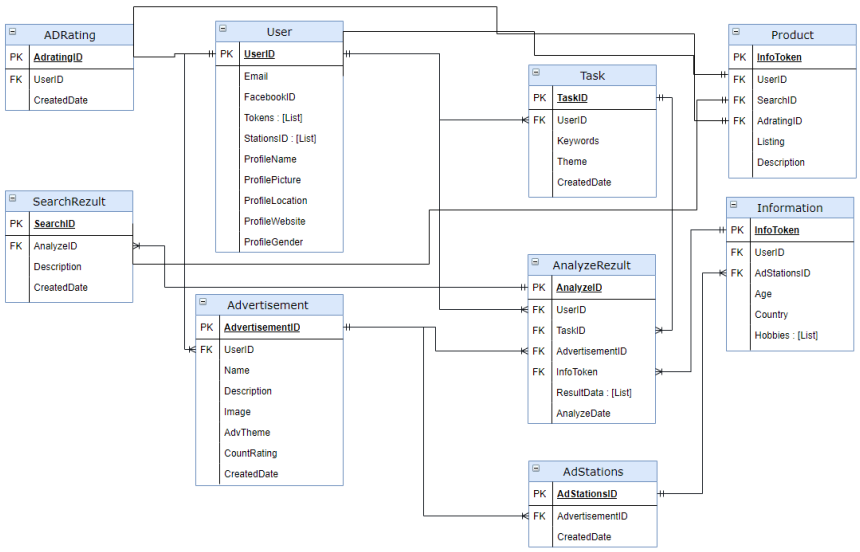


Рисунок 4.1 — Схема бази даних

Таблиця 4.1 — Опис атрибутів базу даних

Тип сутності	Атрибут	Опис	Тип даних	Обмеження	Допустиміть Null
1	2	3	4	5	6
User	UserID	Унікальний ідентифікатор користувача в системі	Number	PK	Ні

	Email	Електронна адреса користувача	String		Ні
	FacebookID	Унікальний ідентифікатор користувача в соціальній мережі Facebook	Number		Ні
User	Tokens	Масив токенів, необхідний для успішної авторизації через соціальну мережу	Array		Ні
	StationsID	Масив реклами, доданих користувачем	Array		Так
	ProfileName	Повне ім’я користувача	String		Ні
	ProfilePicture	Зображення користувача	String		Так
	ProfileLocation	Місце проживання користувача	String		Так
	ProfileWebsite	Веб-сайт	String		Так
	ProfileGender	Стать	String		Так
1	2	3	4	5	6
Search Rezult	SearchID	Унікальний ідентифікатор пошуку в системі	Number	PK	Ні
	AnalyzeID	Унікальний ідентифікатор результатів аналізу	Number	FK	Ні
	Description	Стислий опис результатів пошуку	String		Ні

Advertise-ment	Advertisement-ID	Унікальний ідентифікатор рекламного майданчика	Number	PK	Ні
	UserID	Унікальний ідентифікатор користувача, який створив цей майданчик	Number	FK	НІ
	Name	Назва рекламного майданчика	String		Ні
	Description	Опис рекламного майданчика	String		Так
	Image	Зображення рекламного майданчика	String		Так
	AdvTheme	Масив тем, для яких підходить даний рекламний майданчик	Array		Ні
1	2	3	4	5	6
Advertise-ment	CountRating	Кількість голосів у системі рейтингу майданчиків	Number		Так
	CreatedDate	Дата створення рекламного майданчика	Date		Ні
Rating	RatingID	Унікальний ідентифікатор завдання користувача	Number	PK	Ні
	UserID	Унікальний ідентифікатор користувача, який	Number	FK	Ні

		створює завдання для аналізу			
	Assessment	Масив чисел необхідних для оцінювання	Number		Ні
	Point	Балл, який отримує майданчик	Number		Ні
	CreatedDate	Дата	Date		Ні
Analyze Rezult	AnalyzeID	Унікальний ідентифікатор аналізу у системі	Number	PK	Ні
	UserID	Унікальний ідентифікатор користувача, який проводить аналіз	Number	FK	Ні
1	2	3	4	5	6
Analyze Rezult	TaskID	Унікальний ідентифікатор задання, яке аналізується	Number	FK	Ні
	AdvertisementID	Унікальний ідентифікатор рекламного майданчика, який аналізується	Number	FK	Ні
	InfoToken	Унікальний ідентифікатор списку відомостей про аудиторію	Number	FK	Ні
	ResultData	Масив результатів, отриманих після аналізу	Array		Ні

	AnalyzeDate	Дата проведення аналізу	Date		Ні
AdSta-tions	AdStationsID	Унікальний ідентифікатор біржи реклами	Number		Ні
	Advertisement ID	Унікальний ідентифікатор оголошення для розміщення на власному майданчику	Number		Ні
	CreatedDate	Дата додавання майданчика на біржу	Date		Ні
1	2	3	4	5	6
Product	InfoToken	Унікальний ідентифікатор списку відомостей про інформацію користувачів	Number	PK	Ні
	UserID	Унікальний ідентифікатор користувача для якого йде сбір даних	Number	FK	Ні
	Age	Вік цільової аудиторії	Number		Ні
	Country	Місце перебування цільової аудиторії	String		Ні
	Hobbies	Масив із захопленнями цільової аудиторії	Array		Ні

Як мінімум один аспект розроблюваної системи робить базу даних документно-орієнтованою. В описі зазначено, що атрибут може бути масивам. На ER діаграмі це зазначено як [List]. Документ представлений як один єдиний об’єкт JSON. Нижче у таблиці 4.2 наведені приклади кількох документів.

Таблиця 4.2 — Приклад документів системи

Колекція	Об’єкт
User	{“_id”: “1”, “UserID”: “1”, “Email”: “mamontov.work@gmail.com", “FacebookID”: “167453389”, “Tokens”: [“aXAasdmWWq677sCgRadyhg…”, “XitcWqwpbWyCwU23N:jas…”],
Group	“GroupIDs”: [“12345”, “234582”, “21458”], “ProfileName”: “Yurii Mamontov”, “ProfilePicture”: “/images/123456789.png”, “ProfileLocation”: “Kharkiv, Ukraine”, “ProfileWebsite”: “statcoin.ru", “ProfileGender”: “Male” }

Advertisement	{
	“_id”: “1”,
	“UserID”: “1”,
	“Name”: “Instagranm",
	“Description”: “Best of the best Instagram account”,
	“Image”: “logo/images/123456789_logo.png”,
	“AdvTheme”: [“125”, “2”, “255”],
	“CountRating”: “6,4”,
	“CreatedDate”: “31 May 2018”,
	}

5 МАТЕМАТИЧНИЙ ОПИС ЗАДАЧІ

5.1 Формування рекомендацій гібридним методом коллаборативної фільтрації

Приведемо основні переваги цього підходу, виділені в цій роботі та багатьох інших. Першим в списку переваг підходу найчастіше називають те, що системи, грунтовані на коллаборативної фільтрації, враховують тільки оцінки, виставлені користувачами, і не вимагають ніякої іншої інформації про користувача і продукти. На наш погляд, це швидше недолік, чим перевага підходу. Такий метод рекомендацій є чисто статистичним і, фактично, не враховують персональні переваги конкретного користувача. Відмітимо, що пропонований нижче метод коллаборативної фільтрації не має описаної "переваги". Другим, дійсно важливим, на наш погляд, перевагою підходу є можливість обліку при оцінці переваг не лише даного користувача, але також переваг користувачів з схожими інтересами. Звідси витікає ще одно значима перевага цього підходу, а саме можливість рекомендувати щось зовсім нове, але, можливо, цікаве для користувача (англ. serendipity), використовуючи інформацію про переваги схожих користувачів [6].

Слід зазначити, що рекомендаційні системи, грунтовані на коллаборативної фільтрації, за даними, у загальному випадку, показують більш високу точність, ніж системи, грунтовані на фільтрації контенту. Можливо, це пов'язано з тим, що більшість розроблених раніше рекомендаційних систем на основі фільтрації контенту не враховують повною мірою семантику інтересів користувача.

До основних недоліків систем, грунтованих на коллаборативної фільтрації, можна віднести проблему розрідженості даних : кількість рейтингів, які відомі рекомендаційній системі завжди значно менше, ніж загальна кількість продуктів, рейтинги яких потрібно буде передбачити. Чим менше користувачів оцінило продукт, тим менше вірогідності того факту, що цей продукт буде комусь ще порекомендований, навіть якщо наявні рейтинги високі. Такі системи, на відміну від систем фільтрації контенту, не здатні давати рекомендації відносно нових продуктів,

оскільки для таких продуктів взагалі відсутня інформація про рейтинги інших користувачів. Системи, коллаборативної фільтрації, так само як і системи фільтрації контенту погано вирішують "проблему нового користувача". Окрім цього, до неї додається і проблема "білої ворони" – так говорять про ситуацію, коли інтереси користувача не співпадають з інтересами інших користувачів рекомендаційної системи, отже, в системі не виявляється користувачів, чиї рейтинги можна використати при виробленні рекомендацій.

Відмітимо, що описаний нижче підхід позбавлений більшості описаних недоліків, завдяки використанню семантичного профілю інтересів користувача і семантичної метрики схожості користувачів, що обчислюється на основі цього профілю. Завдяки цьому, підхід швидше можна віднести до гібридних методів вироблення рекомендацій з використанням в основі методу коллаборативної фільтрації.

Розглянемо формальну постановку завдання вироблення рекомендацій методом коллаборативної фільтрації.

Нехай для безлічі користувачів $U_1, \ldots, U_e$ рекомендаційної системи (наприклад, користувачів, чиї оцінки є присутніми в даних Amazon) на основі даних про оцінені ними продукти із залученням додаткової інформації з глобальної бази знань DBpedia побудовані семантичні профілі інтересів користувачів P_r ,..., P_r. При цьому кожному 1 е правилу в профілях поставлено у відповідність значення метрики оцінки "сили" причинного зв'язку $\mu(P^i, \omega)$ [8].

За допомогою алгоритму вироблення рекомендацій необхідно передбачити оцінку (рейтинг) ω_{U_t, I_t} , яку цільовий користувач U_t з безлічі користувачів $U_1, \ldots, U_e$ оцінить новий для нього продукту I_t на основі оцінок $\omega_1, \ldots, \omega_e$, які поставили продукту I_t інші користувачі з безлічі $U_1, \ldots, U_e$.

5.2 Формування рекомендацій колоборативною фільтрацією

Найбільш простим і природним способом вироблення рекомендацій на основі семантичного профілю інтересів користувача.

До переваг методу фільтрації контенту можна віднести "прозорість" механізму рекомендацій - рекомендаційна система, грунтована на фільтрації контенту, завжди може дати пояснення своїм рекомендаціям, тобто продемонструвати користувачеві ті властивості рекомендованого продукту, які відповідають його інтересам. На відміну від методів коллаборативної фільтрації, рекомендаційній системі на основі фільтрації контенту необхідна інформація, що відноситься тільки до одного користувача, що дозволяє зберегти конфіденційність інформації об усіх користувачів рекомендаційній системи. І нарешті, такі рекомендаційні системи здатні рекомендувати користувачем нові продукти - для вироблення рекомендацій не потрібно інформацію про те, які оцінки цьому продукту поставили інші користувачі, оскільки рекомендації грунтуються тільки на властивостях продукту.

Розглянемо детальніше розроблений алгоритм вироблення рекомендацій методом фільтрації контенту на основі семантичного профілю користувача за допомогою асоціативно-причинної класифікації.

Нехай для деякого користувача U_t (користувача, чиї оцінки представлені в наборі даних проекту) рекомендаційної системи на основі даних про оцінені їм продукти $I = \{I_1 ., I_z\}$, із залученням додаткової інформації з глобальної бази знань побудований семантичний профіль інтересів U_t P_r. Цей профіль містить бінарне дерево рішень з присвоєними кожному вузлу наборами асоціативно-причинних правил класифікації. При цьому кожному правилу поставлено у відповідність значення метрики оцінки "сили" причинного зв'язку $\mu(P^i, \omega)$.

За допомогою алгоритму вироблення рекомендацій необхідно передбачити оцінку (рейтинг) $\omega_{U,I}$, яку користувач U_t поставить новому для нього продукту I_t на основі профілю і характеристик продукту I_t.

Далі необхідно вибрати ті правила профілю користувача, які працюють на продукті I_t. Опишемо процедуру вибору правил, яка повинна виконуватися для кожного вузла бінарного дерева ухвалення рішень.

Нехай в деякому вузлі є безліч предикатів виду $P_k^i(X_k^i)$, а для продукту I_t витягнуті значення усіх його атрибутів X^j, $j=1,..,m$.

1) Якщо усі предикати $P_k^i(X_k^i)$ ~ (посилкА правил класифікації) у вузлі ухвалення рішень розглянуті, то перехід до п. 11, інакше до п. 2.

2) Виберемо наступний не переглянутий предикат $P_k^i(X_k^i)$ зі списку правил.

3) Якщо усі атрибути продукту I_t переглянуті, то виконується перехід до п. 1, інакше – до п. 4.

4) Виберемо наступний атрибут X^j зі списку атрибутів продукту I_t.

5) Якщо вибраний атрибут X^j відповідає атрибуту і X^i з профілю користувача, для якого побудований предикат $P_k^i(X_k^i)$, то виконується перехід до п.4, інакше – до п.2.

6) Якщо усі значення $x_y^i \epsilon X_k^i$ переглянуті, то п. 1, інакше п. 7.

7) Вибирається наступне не переглянуте значення $x_y^i \epsilon X_k^i$ (якщо атрибут X^i цілочисельний, то $x_y^i \epsilon X_k^i$ буде інтервалом значень атрибуту X^i, отриманий в ході попередньої дискретизації усіх значень атрибуту X^i).

8) Якщо значення $x_t^j = x_y^i$ (чи $x_t^j \epsilon x_y^i$, якщо x_y^i– це деякий інтервал значень з X_k^i), де x_t^j - це значення атрибуту X^j для продукту I_t, то виконується перехід до п. 9, інакше – до п. 1.

9) Додається правило, що відповідає предикату $P_k^i(X_k^i)$, в список правил I_t, що спрацювали для продукту.

10) Перехід до п. 1.

11) Кінець.

5.3 Кластеризація користувачів

Для підвищення ефективності процедури вироблення рекомендацій необхідно виконати попередню кластеризацію користувачів на основі схожості їх інтересів. Такі

кластери можуть бути такими, що перетинаються, тобто один користувач може належати до декількох кластерів. Це відбиває реальний стан речей.

Після виконання попередньої кластеризації користувачів для обчислення оцінки $\omega_{U,I}$ продукту I_t для користувача U_t будуть використані тільки оцінки $\omega_1,...,\omega_c$, виставлені цьому продукту користувачами $U_1, ..., U_c$, котрі належать тому ж кластеру, що і користувач U_t.

Перш ніж застосовувати алгоритм кластеризації користувачів, необхідно нормувати їх профілі. Виконується це таким чином. Для кожного користувача $U_1, ..., U_c$ його інтереси, представлені предикатами і правив з вузлів дерева рішень нижнього рівня, упорядковуються по убуванню значення метрики причинного зв'язку $\mu(P^i, \omega)$; послідовно обчислюється поточна сума значень метрики по порядку дотримання інтересів і доки поточна сума не досягла значення рівного значення, поточних інтересів додаються в підсумковий нормований профіль, усі подальші - відсікаються.

На основі нормованого профілю користувача для кожної пари користувачів з $U_1, ..., U_c$ можна розрахувати значення міри схожості їх інтересів. Як міра схожості пари користувачів використовується міра Танимото, яка має наступний вигляд:

$$Sim'(U_k, U_l) = \frac{\boldsymbol{Pr}'_{U_k} \cap \boldsymbol{Pr}'_{U_l}}{\boldsymbol{Pr}'_{U_k} + \boldsymbol{Pr}'_{U_l}} \qquad (4.1)$$

де $\boldsymbol{Pr}'$ і $\boldsymbol{Pr}'$ кількість інтересів в нормованих профілях користувачів U_k і U_l відповідно; $\boldsymbol{Pr}' \cap \boldsymbol{Pr}'$ - кількість загальних інтересів користувачів U_k і U_l. При цьому інтерес вважається загальним для користувачів U_k і U_l у тому випадку, коли у обох користувачів в нормованому профілі є правила, правила яких побудовані для одного і того ж атрибута, і ці правила мають в укладенні одну і ту ж мітку класу.

5.4 Алгоритм обчислення рейтингу

Для обчислення оцінки $\omega_{U,I}$ продукту I_t для користувача U_t спочатку необхідно вибрати усіх користувачів U_i, які знаходяться в одному декількох кластерах с цільовим користувачем U_t. Тільки оцінки цих користувачів (при їх наявності) далі використовуються при підрахунку значення оцінки продукту I_t. Далі з цих користувачів вибираються ті, які мають оцінку для продукту I_t. Нехай ці користувачі утворюють безліч U_t . Пропонується два основні варіанти обчислення рейтингу $\omega_{U,I}$:

$$\omega_{U,I_t} = k \sum_{U_i \in U^t} \left(sim(U_t, U_i) \times \left(\omega_{U,I} \right)_{i} \right)_{t} \quad (4.2)$$

$$\omega_{U,I_t} = \hat{\omega}_{U_t} + k \sum_{U_i \in U^t} \left(sim(U_t, U_i) \times \left(\omega_{U,I_t} - \hat{\omega}_{U_t} \right)_{i} \right) \quad (4.3)$$

Де метрика схожості користувачів U_t *і* U_i; k - це коефіцієнт нормалізації.

Вираз (4.2) дозволяє вичислити значення $\omega_{U,I}$ як зважене середнє значень $\omega_{U,I}$. При цьому вагою виступає значення міри схожості $sim(U_t, U_i)$ між користувачами U_t *і* U_i: чим більше значення $sim(U_t, U_i)$, тим більший вклад в значення $\omega_{U,I}$ вносить рейтинг $\omega_{U,I}$ користувача U_i. Слід зазначити важливий недолік цього підходу: різні користувачі можуть по-різному (суб'єктивно) використати шкалу рейтингів. Для деякого користувача оцінка 4 з 5 може означати, що йому дуже сподобався продукт, а для іншого 4 - це посередня оцінка продукту. Здолати описаний недолік дозволяє формула (4.3), яка використовує скоректоване зважене середнє значення рейтингів : замість абсолютного значення рейтингу $\omega_{U,I}$ користувача U_i при підрахунку середнього значення рейтингу використовується його відхилення від середнього значення рейтингу користувача Ui [13].

6 РОЗРОБКА ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ

6.1 Загальні відомості

Під розробленою системою слід представляти веб-сервіс, з допомогою якого майбутні або діючи підприємці мають змогу проводити аналіз цільової аудиторії. Також вони мають змогу переглядати результати аналізу у вигляді списків, таблиць та графіків. Після проведення аналізу користувачу пропонуються місця для розміщення реклами, виходячи з теми та ключовими словами. Власники рекламний майданчиків, наприклад блогери, мають змогу додати до системи власний майданчик для подальшої маркетингової співпраці з підприємцями. Керування завданнями, біржою реклами виконується в особистому кабінеті. Також самою головною функцією є отримання рекомендацій, де користувач отримує майданчики на основі своїх вподобань та досвіді колег.

6.2 Опис логічної структури ПО

6.2.1 Опис та вибір архітектури системи

Архітектура програмного забезпечення — один із способів, який допомагає структурувати програмну систему, тобто це розділ всієї системи на абстракції. Для кожної потреби ти технології існують різні типи: «Модель-вид-контролер, також відомий як MVC», «Сервісно-орієнтована архітектура (SOA)», «Трирівнева архітектура (Multitier architecture)» та «Клієнт-сервісна архітектура» [?]. Для програмної системи найкраще підходить клієнт-серверна архітектура, яка зображена на рисунку 6.1.

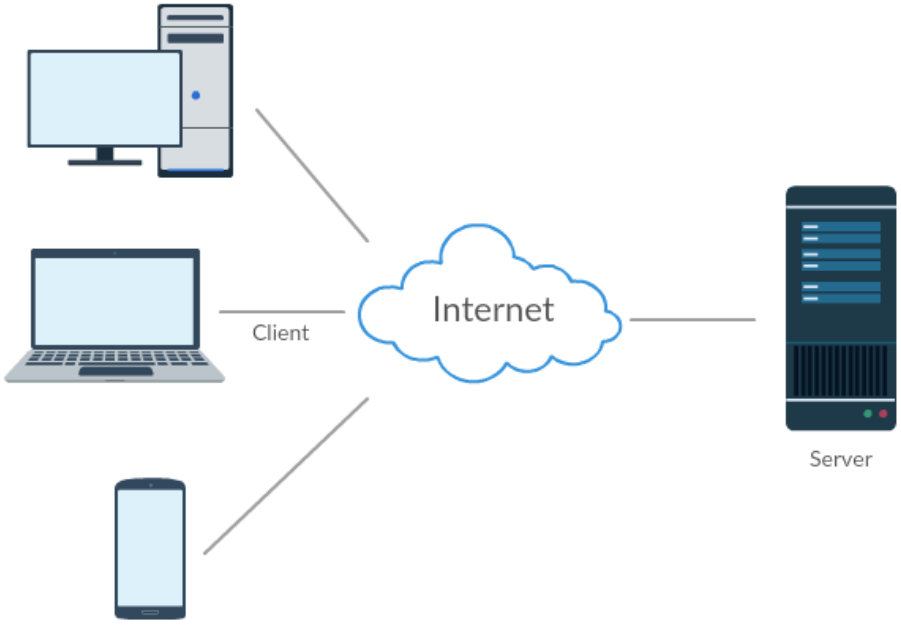


Рисунок 6.1 — Схема клієнт-серверної архітектури

Клієнт-сервісну архітектуру можна вважати базовою для розробки веб-додатків, проте є великий недолік, архітектура не розрахована на велике навантаження. Для начальних етапів потужності вистачить, але варто постійно тримати зворотній зв’язок з користувачами щоб знати про їх зацікавленість у проекті. Якщо продукт якісний та аудиторія проявляє зацікавленість, потік користувачів буде тільки збільшуватися і для цього буде необхідно використовувати більш складну архітектуру, тому під час розробки системи необхідно дотримуватися модульного стилю написання коду. При правильному підході, у випадках високих навантажень на сервер, архітектуру програмного засобу можна буде замінити, наприклад, на сервісну.

6.2.2 Вибір мови програмування

Сьогодні, на ринку представлена дуже велика кількість мов програмування. За різними оцінками, в світі налічується 400-600 production-ready мов програмування. І, як мінімум, на два порядки більше експериментальних проектів. Для розробки сервісу аналізу знадобиться мова, яка дозволяє використовувати клієнт-сервісну архітектуру. Популярними представниками є : Java, C#, JavaScript. Розглянемо кожну з них для отримання оптимального рішення для розроблюваної системи:

– Java — поширена та кросплатформенна мова програмування, тобто продукт, який розробляється з допомогою Java, може працювати більш ніж на одній програмній операційній системі або апаратній платформі. Особливістю кросплатформеності можна вважати суттєве скорочення витрат на розробку продукту.

– C# — мова програмування розроблена корпорацією Microsoft, яка безпосередньо націлена на платформу Windows. Корпорація працює над тим, щоб програмні продукти, розроблені цією мовою програмування, мали можливість бути перенесені на інші операційні системи, тобто компанія націлена на те, щоб зробити C# кросплатформенною мовою програмування. Особливістю мови є охоплення додатків будь-якого типу: консольні, десктопні та веб-додатки, сервіси та мобільні застосунки.

– JavaScript — мова програмування з самого початку створена для розробки клієнтської частини у веб-додатках [?]. Час не стоїть на місці і в слід тенденціям, у 2009 році Раяном Даром створюється платформа, яка отримала назву Node.js. І тепер мова, яка використовувалася виключно в браузерах, стала мовою загального використання з великою спільнотою. Особливістю мови можна виділити низький поріг входження, тобто мова легка у вивченні, наприклад, мінімально життєвий продукт (MVP) може бути створити одна людина за лічені дні.

Для розробки програмних продуктів можна використовувати будь-яку, кожна мова має свої особливості, але за рахунок складності Java та C#, прийнято рішення почати розробку на мові JavaScript [16].

6.2.3 Обрані програмні засоби

Перед розробкою необхідно визначитися з програмними засобами. Для цього було обрано IDE WebsSorm. С допомогою цього комерційного програмного засобу з’являється можливість побачити усі переваги обраної мови програмування. Для розробки серверної частини програмного продукту обрано — NodeJS. Також обрано фреймворк AngularJS, з допомогою якого швидкість розробки клієнтської частини веб-додатку значно збільшується, при цьому якість програмного засобу не втрачається.

6.3 Вхідні та вихідні дані для розроблюваного програмного засобу

Для того щоб аналізувати вхідні дані необхідно розглянути дані, якими в процесі роботи с сервісом використовує кожний окремий користувач. Цими даними можуть виступати завдання для аналізу або дані про рекламний майданчик. Розглянемо детально.

Перед користуванням сервісу, користувачу мережі Інтернет необхідно зареєструватися у системі. Для комфорту було вирішено створити можливість реєстрації через соціальну мережу Facebook (FB), надалі це єдиний спосіб реєстрації. Все проходить в автоматичному режимі: на головному вікні веб-сервісу користувач натискає на кнопку «Реєстрація» та надає права для програмного засобу, демонстрацію роботи реєстрації зображено на рис. 6.2. База даних зберігає дані, які були отримані через API у таблицю User (розділ 5.2), через це при кожному наступному використанню розроблюваної системи треба лише авторизуватися.

Після авторизації у користувача є два сценарію продовження дій. Все залежить від потреб та мети користувачів: зробити аналіз цільових груп користувачів, додати власний майданчик, наприклад акаунт соціальної мережі Instagram, також користувач, знаючи цільову аудиторію свого бізнесу, може переглянути на біржі реклами майданчики для розміщення, та на основі переглянутих варіантів може отримати рекомендації. Розглянемо головний варіант використання, а саме рекомендаційну частину.

– користувач мережі використовує розроблювану систему, щоб отримати дані про цільову аудиторію, при цьому, після проходження аналізу, користувач має змогу переглянути актуальні рекламні майданчики запропоновані системою. Перш за все, у вікні завдання користувач заповняє необхідні поля: ключові слова, тематика. Після цього, виходячи із отриманої інформація через API соціальних мереж, система починає: збір та збагачення даних, сегментування, інтерпретацію сегментів. Вся інформація записується у таблиці AnalyzeResult, Information, Task. Як тільки користувач почне передивлятися варіанти та ставити оцінки система почне активно рекомендувати схожі за тематикою, ЦА, демографіє. Майданчики для розміщення (рис. 6.5).

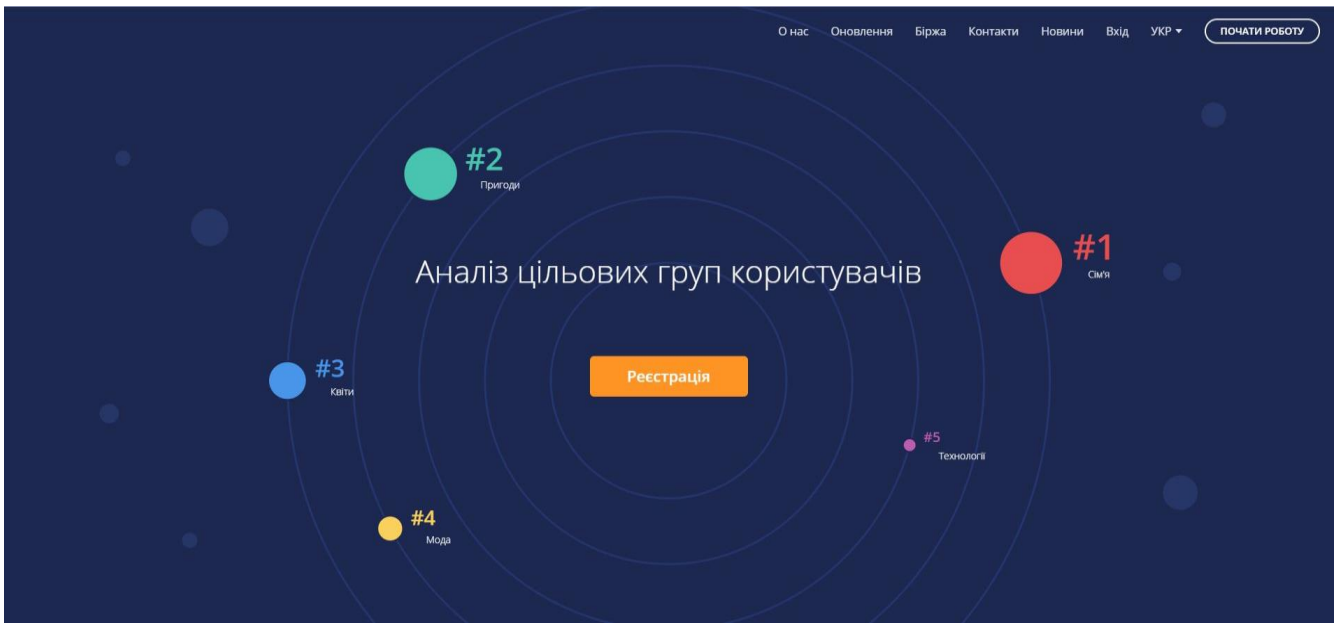


Рисунок 6.2 – Реєстрація через соціальну мережу FB

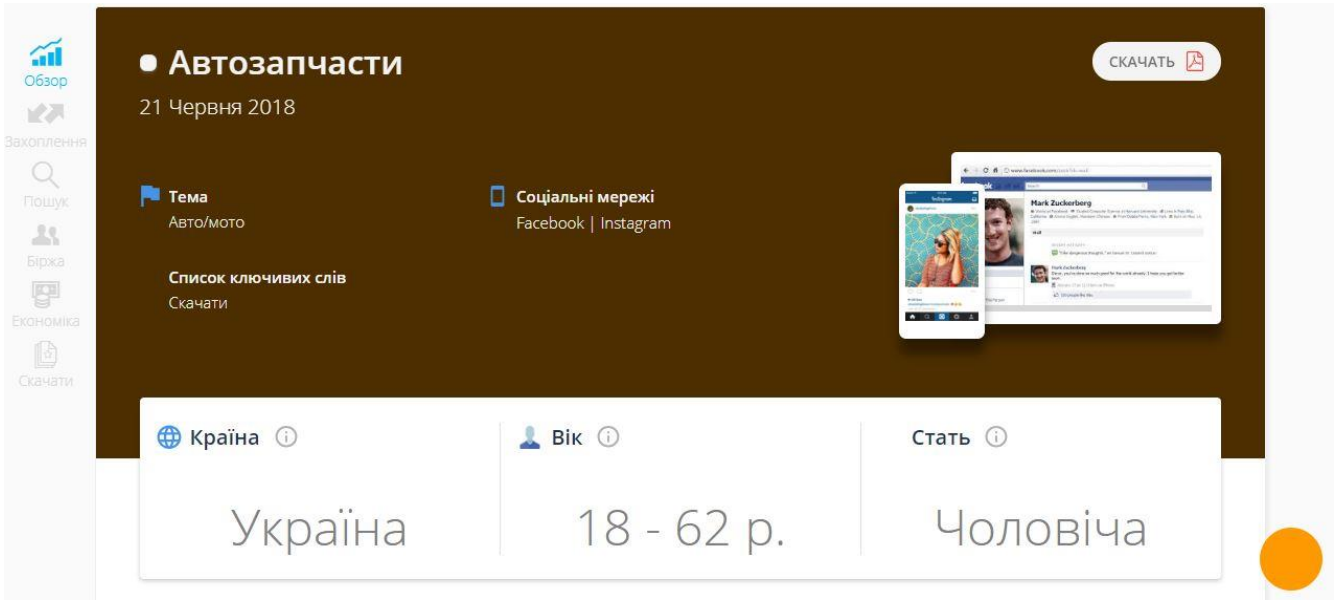


Рисунок 6.3 – Результат аналізу цільових груп користувачів

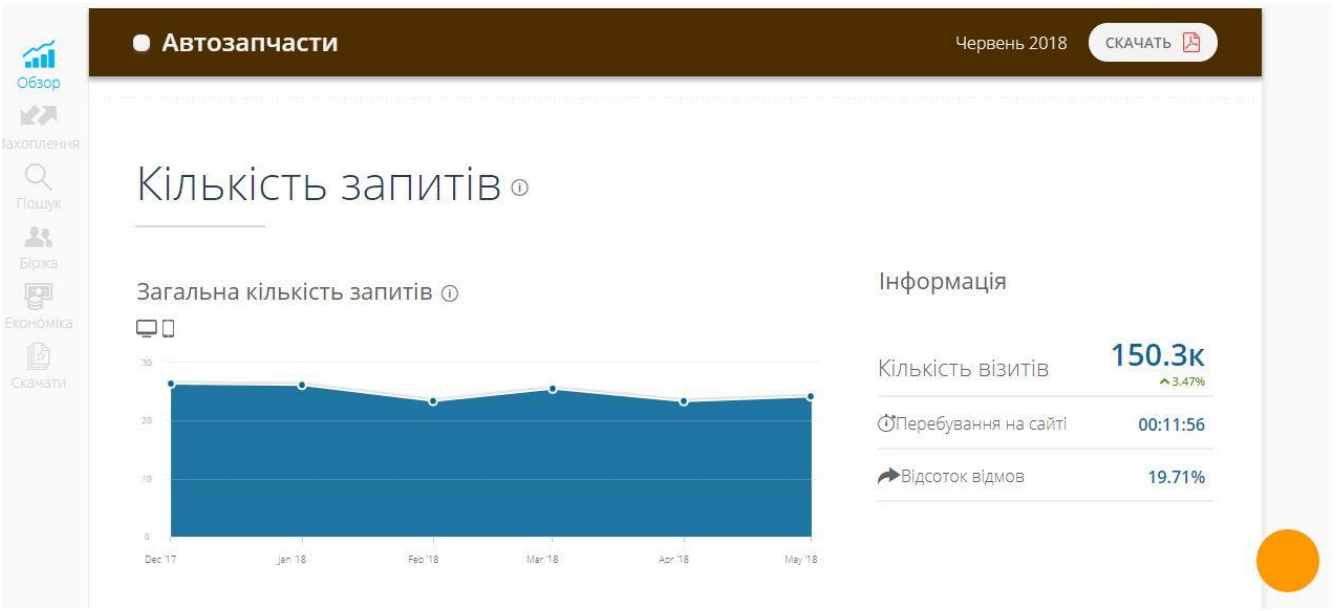


Рисунок 6.4 – Отримання вихідних даних роботи функції

Користувач сервісу використовує систему, щоб отримати рекомендації, які сформовані на основі попередніх пошуках та оцінках. (рис. 6.5).

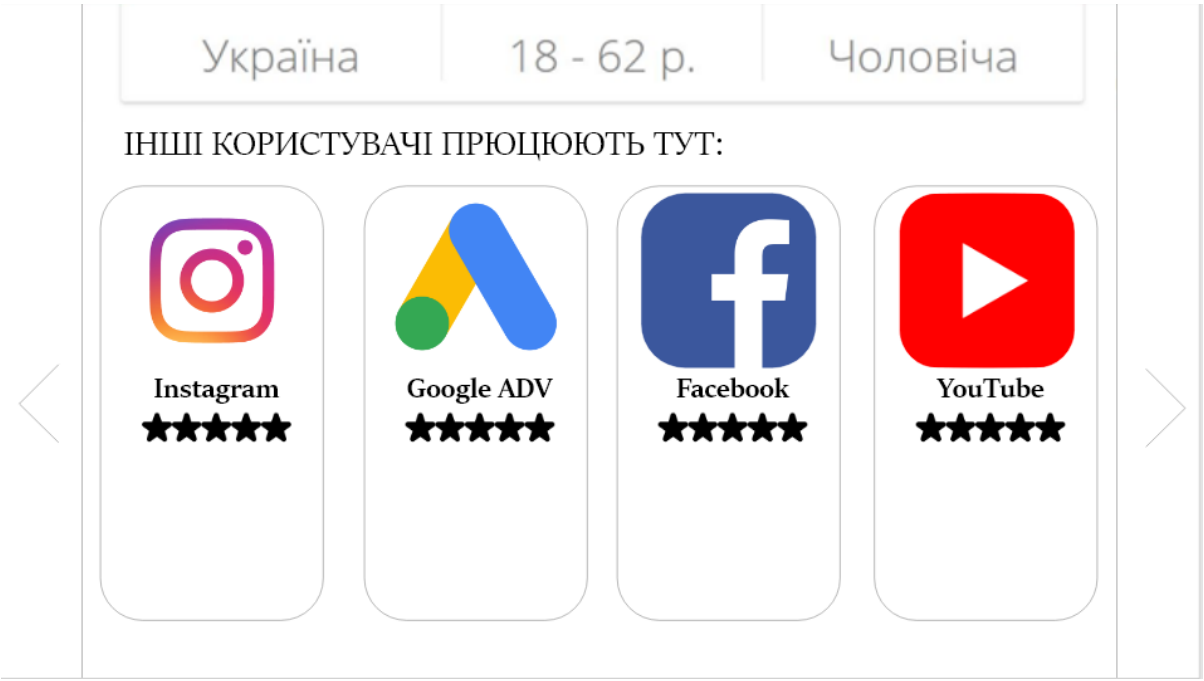


Рисунок 6.5 – Сторінка рекомендації

ВИСНОВКИ

В атестаційній роботі вирішено актуальне наукове завдання - розробка алгоритмів рекомендаційних систем методом колаборативної фільтрації, а також виконано експериментальне дослідження цих алгоритмів для конкретного застосування - рекомендаційної системи для маркетингового сервісу.

Для отримання інформації щодо предметної області та для досягнення поставленої мети, були розглянуті та проаналізовані вже існуючи програмні засоби рекомендаційних систем. Результатом цих досліджень можна вважати сформовану задачу проектування, яка визначає основні бізнес-процеси системи та функції, які система має виконувати.

Для отримання чіткого розуміння про процеси, які відбуваються у системі, було проведено детальний опис об’єкта дослідження за допомогою методології IDEF0. Були розроблені діаграми прецедентів та діаграма діяльності.

Для виконання основної функції дипломної роботи були розглянуті та проаналізовані основні методи та сформовані алгоритми колоборативної фільтрації, визначені їх переваги та недоліки. Серед усіх існуючих методів був алгоритм формування рекомендацій методом фільтрації контенту, вся програмна реалізація цього методу була виконана мовою програмування JavaScript.

В ході написання дипломного проекту були отримані наступні результати:

- Розроблений алгоритм гібридної коллаборативной фільтрації, який використовує семантичну схожість інтересів користувачів, тоді як стандартні підходи використовують тільки статистичну інформацію.
- Запропонований алгоритм вироблення кросс-домених рекомендацій на основі семантичного профілю користувача і коллаборативної фільтрації. Алгоритм дозволяє з необхідною точністю передбачати рейтинги, які деякий користувач присвоїть продуктам з тих доменів, в яких він раніше не виставляв рейтингів.

Програмний продукт, який був розроблений в рамках атестаційної роботи потребує подальшого вдосконалення функціоналу. Основними можна виділити:

локалізація (переклад) сайту на всі популярні мови світу, щоб не тільки українці могли використовувати програмний засіб; для ще більш якісного аналізу необхідно впроваджувати алгоритми машинного навчання та штучного інтелекту; необхідно розробити десктоп-додаток, з метою постійного доступу, навіть без підключення до мережі.

Розроблювана система отримала простий та інтуїтивно зрозумілий інтерфейс, який відповідає тенденціям веб-дизайну.

Користувачі, в яких знання програмування на вищому рівні мають змогу використовувати дане рішення у своїх продуктах, тому що розроблена система має відкриті вихідні коди на порталі GitLab.

ПЕРЕЛІК ПОСИЛАНЬ

1. Методичні вказівки до організації виконання та захисту кваліфікаційної роботи ОКР «магістр» за напрямом 6.050101 – «Комп'ютерні науки» для студентів усіх форм навчання / [упоряд.: І. В. Гребеннік, М. В. Євланов, В. Г. Іванов, Л. М. Ребезюк, Н. В. Рябова]. – Харків : ХНУРЕ, 2016. – 56 с.

2. Гомзин А. Г., Шулік А. В. Системи рекомендацій : огляд сучасних підходів [Електронний ресурс]: праці ИСП РАН. 2012.

3. Мандель И.Д. Кластерний аналіз. М.: Финансы и статистика, 1988. – 176 с.

4. Городецкий В.И. Состояние и перспективы интеллектуального анализа больших данных // Труды всероссийской конференции «Интеллектуальные технологии в управлении», Санкт-Петербург, 7- 9 октября 2014 г. С. 61–73.

5. Han F., Liu H. Transition matrix estimation in high dimensional time series // Proceedings of the 30th International Conference on Machine Learning. USA. 2013. Vol. 28. pp. 172–180.

6. Lops P., De Gemmis M., Semeraro G. Content-based recommender sys-tems: state of the art and trends. In: Ricci F., Rokach L., Shapira B. (eds.) Recommender systems handbook, pp. 73-105. Springer, Hedelberg. 2011. pp.73-106.

7. Desrosiers C., Karypis G. A comprehensive survey of neighborhood-based recommendation methods. In: Ricci F., Rokach L., Shapira B. (eds.) Recommender systems handbook. Springer, Hedelberg. 2011. pp. 107-144.

8. Гвоздева В. А. Основы построения автоматизированных информационных систем / В. А. Гвоздева, И. Ю. Лаврентьева. – М.: ФОРУМ: ИНФРА-М, 2007. – 320 с.

9. Кузнєцов С. Д. Основи баз даних : навчань. посібник / Сергій Дмитрович Кузнєцов. - 2-е видавництво, испр. - М.: Інтернет-університет інформаційних технологій : Біном. Лабораторія знань, 2007. - 484 с.

10. Aliferis C.F., Statnikov A., Tsamardinos I., .Xenofon S.M., Koutsoukos D. Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification. Part I: Algorithms and Empirical Evaluation. // Journal of Machine Learning Research. 2010. No. 11. pp. 171-234.

11. Лекція в Гугл [Електронний ресурс]: Як працюють рекомендаційні системи. - Електрон. текст. дан. - Режим доступу : http://habrahabr.ru/company/google/blog/241455/

12. Филонова В.А. Разработка модели персонализации и алгоритма управления контентом веб-сайта с учетом постоянных и текущих потребностей пользователя // Портал магистров ДонНТУ, 2014.

13. Дьяконов А.Г. Алгоритмы для рекомендательной системы: технология LENKOR // Бизнес-информатика . 2012.

14. Михнюк Д.В. Разработка и исследование алгоритмического обеспечения интеллектуальной системы формирования рекомендаций на основе методов коллаборативной фильтрации, 2014.

15. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item–based collaborative filtering recommendation algorithms. In Proceedings of the Tenth International World Wide Web Conference pp. 285–295 (2001)

16. Sahar S., Mansour Y. An empirical evaluation of objective interestingness criteria // Proceeedings of SPIE Conference on Data Mining and Knowledge Discovery, Orlando, FL. 1999. pp. 63-74.

17. Wikipedia.org: the free encyclopedia. Precision and recall // URL: http://en.wikipedia.org/wiki/Precision_and_recall

18. Weka 3: Data Mining Software. URL: http://www.cs.waikato.ac.nz/ml/weka