

УДК 519.854.2

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

АТЕСТАЦІЙНА РОБОТА Пояснювальна записка

Рівень вищої освіти другий (магістерський)

ГЮИК 50 2810.012 ПЗ

Розробка та дослідження рекомендаційних систем в готельній галузі

за допомогою Web-технологій
(тема)

Виконав:

Студент 2 курсу, групи СПРм-19-1

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

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

Освітня програма Системне проектування

(повна назва освітньої програми)

Столбовий А. Є.

(прізвище, ініціали)

Керівник доц. Петрова Р.В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Гребеннік І. В.

(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

Рівень вищої освіти другий (магістерський)

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

Освітня програма Системне проектування
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Столбовому Артему Євгеновичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та дослідження рекомендаційних систем в готельній галузі за допомогою Web-технологій

затверджена наказом по університету від «02» листопада 2020 року №1516Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 18 грудня 2020 р.

3. Вихідні дані до роботи Науково-технічні публікації, дані статей, дані проектів щодо рекомендаційних систем, результати експериментальних досліджень

4. Перелік питань, що потрібно опрацювати в роботі Вступ, 1. Огляд і аналіз сучасного стану проблеми, що розглядається, 1.1 Поняття рекомендаційної системи на основі спільної фільтрації, 1.2 Рекомендації системи на основі спільної фільтрації, 1.3 Рекомендаційні системи на основі вмісту, 1.4 Рекомендаційні системи засновані на знаннях, 2. Рекомендаційна система для сервісів зі складними предметними сферами, 2.1 Стратегія надання рекомендацій, 2.1 Стратегія надання рекомендацій, 2.2 Математичний опис задачі, 3. Постановка задачі та наліз предметної області, 3.1 Формалізована постановка задачі, 3.2 Вимоги до системи, яку ми розробляємо, 3.3 Характеристика предметної області, 3.4 Огляд і аналіз існуючих аналогів, 3.5 Специфікація вимог до задачі, 4. Проектування системи, 4.1 Опис системних вимог, 4.2 Опис функціональних вимог, 4.3 Опис архітектури системи, 5. Розробка інформаційного забезпечення, 5.1 Обґрунтування вибору СУБД, 5.2 Опис розробленої БД, 6. Розробка

програмного забезпечення та експериментальний аналіз алгоритму створення, 6.1 Вибір мови програмування, 6.2 Опис алгоритмів інформаційної системи, 6.3 Експериментальний аналіз та наочні результати роботи РС, 6.4 Тестування розробленого програмного забезпечення, Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

3.12 Діаграма варіантів використання 4.1 Концептуальна діаграма РС, 4.2 Діаграма декомпозиції, що представляє функції системи, 4.3 Діаграма декомпозиції, що представляє функції реєстрації, 4.4 Діаграма декомпозиції, роботи користувача з сайтом, 4.5 Декомпозиція функції декомпозивання, 6.1 Схема алгоритму бронювання номеру, 6.2 Схема алгоритму пошуку номеру, 6.3 Схема алгоритму роботи рекомендаційної системи

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Розділи спеціальної	доц. Петрова Р.В.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	02.10.2020	
2	Аналіз завдання, підбір літератури	01.11.20-05.11.20	
3	Аналіз літератури з досліджуваної проблеми	05.11.20-07.11.20	
4	Аналіз технічних засобів	08.11.20-10.11.20	
5	Розробка методу	10.11.20-25.11.20	
6	Програмна реалізація	25.11.20-29.11.20	
7	Оформлення пояснювальної записки	29.11.20-1.12.20	
8	Перевірка на плагіат	10.12.20	
9	Рецензування	10.12.20	
10	Підготовка презентації та доповіді	16.12.20	
11	Занесення роботи в електронний архів	17.12.20	
12	Попередній захист атестаційної роботи	18.12.20	

Дата видачі завдання 23 жовтня 2020 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Петрова Р. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської атестаційної роботи: 83 с., 3 табл., 37 рис., 2 додатки, 27 джерел. Графічна частина атестаційної роботи містить: 7 плакатів.

РЕКОМЕНДАЦІЙНІ СИСТЕМИ, CONTENT-BASED FILTERING, COLLABORATIVE FILTERING, KNOWLEDGE BASED RECOMMENDERS

Об'єктом дослідження атестаційної роботи є веб-сервіси, що спеціалізуються на наданні кінцевому користувачеві деякого результату (продукту) та охоплюють предметні області, простір елементів яких є складним, з точки зору можливості отримання оцінок та має низьку частоту переглядів. З огляду на сучасний рівень розвитку інформаційних систем, впровадження рекомендаційних алгоритмів до існуючих веб-систем має велику актуальність. Рекомендаційні алгоритми дозволяють заощаджувати велику кількість часу користувачам систем, а власникам утримувати клієнтів на ресурсі та допомагати їм прийняти рішення.

Предмет дослідження – алгоритми та методи створення рекомендацій, які застосовуються для прогнозування уподобань користувачів та надання їм пропозицій, згідно їх уподобань.

Мета атестаційної роботи – дослідження існуючих методів побудови рекомендацій на предмет можливості їх використання у сферах.

Досягнення мети атестаційної роботи базується на комплексному використанні методів теорії фільтрації інформації, що стосуються створення рекомендаційних систем різних видів

Результат атестаційної роботи – стратегія надання рекомендацій, рекомендаційна система яка використовує розроблену стратегію, а також веб-сервіс з пошуку номерів, з вбудованою до нього рекомендаційною системою.

ABSTRACT

Explanatory note: 83 pages, 3 tables, 27 figures, 2 applications, 27 sources.
Graphic material 7 p.

RECOMMENDER SYSTEMS, CONTENT-BASED FILTERING,
COLLABORATIVE FILTERING, KNOWLEDGE BASED RECOMMENDERS

The object of the work is web services that specialize in providing end-user access to certain content, covering subject areas of which the space of the elements is complex, in terms of being able to receive ratings and having a low frequency of views. I will look at the current level of development of information systems, the introduction of recommendation algorithms to the current web systems is of great relevance. The recommendation algorithms allow for a great number of hours for the systems and for the owners to use the clients on the resources and for additional assistance and acceptance of the solution.

The subject of the work is the algorithms and methods of creating recommendations, which are used to predict users' preferences and to provide offers according to their preferences.

The purpose of the work is to investigate existing methods of constructing recommendations for the possibility of their use in areas.

Methods of development – methods of information filtering theory related to the creation of recommendation systems.

The results of the work – strategy for making recommendations, the recommender system based on it, hotel number search service which uses a developed recommendation system.

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	9
ВСТУП.....	10
1 ОГЛЯД І АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ, ЩО РОЗГЛЯДАЄТЬСЯ.....	14
1.1 Поняття рекомендаційної системи.....	14
1.2 Рекомендаційні системи на основі спільної фільтрації.....	17
1.3 Рекомендаційні системи на основі вмісту.....	23
1.4 Рекомендаційні системи засновані на названнях.....	26
2 РЕКОМЕНДАЦІЙНА СИСТЕМА ДЛЯ СЕРВІСІВ ЗІ «СКЛАДНИМИ» ПРЕДМЕТНИМИ СФЕРАМИ.....	30
2.1 Стратегія надання рекомендацій.....	29
2.2 Математичний опис задачі.....	32
3 ПОСТАНОВКА ЗАДАЧІ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	38
3.1 Формалізована постановка задачі.....	38
3.2 Вимоги до системи, яку ми розробляємо.....	39
3.3 Характеристика предметної області.....	39
3.4 Огляд і аналіз існуючих аналогів.....	40
3.5 Специфікація вимог до задачі.....	48
4 ПРОЕКТУВАННЯ СИСТЕМИ.....	52
4.1 Опис системних вимог.....	52
4.2 Опис функціональних вимог.....	53
4.3 Опис архітектури системи.....	58
5 РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ.....	60
5.1 Обґрунтування вибору СУБД.....	60
5.2 Опис розробленої БД.....	61
6 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЕКСПЕРИМЕНТАЛЬНИЙ АНАЛІЗ АЛГОРИТМУ СТВОРЕННЯ.....	64
6.1 Вибір мови програмування.....	64
6.2 Опис алгоритмів інформаційної системи.....	68
6.3 Експериментальний аналіз та наочні результати роботи РС.....	70
6.4 Тестування розробленого програмного забезпечення.....	74
ВИСНОВКИ.....	79
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	81

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- Collaborative filtering – колаборативна або спільна фільтрація;
Content-based filtering – фільтрація на основі вмісту або контентна фільтрація;
Memorybased – фільтрація заснована на сусідстві; Modelbased – фільтрація заснована на моделі;
MVC –Model-View-Controller – архітектурний шаблон; РС – рекомендаційна система.

ВСТУП

Останнім часом кількість інтернет-платформ, що пропонують користувачам різноманітні товари, збільшується з експоненціальною швидкістю, а рекомендаційні системи стають незамінними компонентами будь-якої подібної веб-системи. Різноманітні інтернет-магазини, електронні дошки оголошень, соціальні мережі, платформи з аудіо- та відео-контентом, комерційні організації, що мають великі каталоги товарів змогли отримати вигоду від подібних систем. Адже орієнтація у великих каталогах товарів надзвичайно складна і потенційного користувача може втомити робота з такою платформою або він не зможе знайти саме ті елементи, які йому потрібні, навіть з урахуванням пошукових фільтрів, а рекомендаційні системи дозволяють вирішити подібні проблеми.

У загальному розумінні рекомендаційні системи — це активні системи фільтрації інформації, які намагаються надати користувачу інформаційні елементи (кіно, телебачення, музика, новини, веб-сторінки), у яких він зацікавлений. На разі існує дві основні стратегії по створенню рекомендаційних систем: спільна (колаборативна) фільтрація та фільтрація на основі контенту. А також гібридні підходи, що використовують обидві стратегії у певній мірі [1, 2, 3]. Зазвичай великі сервіси, такі як Netflix, YouTube чи Amazon, які мають величезні клієнтські бази і каталоги товарів, використовують колаборативну фільтрацію або гібридний підхід, хоча при цьому рекомендаційні системи засновані на контенті також можуть використовуватися.

Колаборативна фільтрація у рекомендаційних системах базується на попередніх взаємодіях між користувачами та елементами, які використовуються для того, щоб створювати нові рекомендації. Такі взаємодії зберігаються у вигляді матриць оцінок «елементи-користувачі». Основна ідея

полягає у тому, що цих попередніх оцінок достатньо для того щоб знаходити схожих користувачів та/або схожі (близькі) елементи та створювати рекомендації на їх основі [1 – 3].

Методи фільтрації на основі вмісту засновані на описі елементів та профілі переваг користувача, що будується на попередніх взаємодіях користувача з елементами. Ця взаємодія зберігається у матрицю оцінок елементів, на основі якої потім будуються уподобання користувача. Основна ідея полягає у тому щоб пропонувати користувачу ті елементи, які найбільше відповідають його персональним уподобанням [1, 2, 3].

Існуючі підходи базуються на різних способах використання будь-чєї історії рейтингів, для створення рекомендацій. Але такої історії уподобань може не бути. Цілком можливо, що у користувача взагалі не буде оцінок і фактично кожен новий користувач буде мати таку проблему. Це називається проблемою холодного запуску, і це реальна проблема в проектуванні рекомендаційних систем, тому що важко робити прогнози, коли уподобання користувача невідомі. Іншою причиною недостатньої кількості оцінок, може бути те, що рекомендовані речі є дуже складними та комплексними. Такі предмети, як машини, нерухомість, ювелірні вироби купуються нечасто і можуть бути на стільки складними, що оцінки для них можуть бути недоступні. Тому створення рекомендаційних систем, для сфер з обмеженою можливістю отримання оцінок є актуальною задачею та предметом активних досліджень.

Метою даної роботи є дослідження існуючих методів побудови рекомендацій на предмет можливості їх використання у сферах, простір елементів яких є складним та/або має низьку частоту переглядів та створення системи рекомендацій, для таких предметних областей.

Поставлена мета роботи обумовила наступні завдання дослідження:

- огляд існуючих стратегій по створенню рекомендаційних систем;

аналіз можливості застосування розглянутих стратегій в рамках сфер що мають складний (комплексний) простір елементів, який використовується нечасто;

- розробка рекомендаційної системи, що може використовуватися для подібних сфер;
- розробка веб-сервісу для однієї з таких предметних областей;
- реалізація та впровадження розробленої рекомендаційної системи до створеного сервісу.

Об'єктом дослідження є веб-сервіси, що спеціалізуються на наданні послуг бронювання номерів.

Предметом дослідження є алгоритми та методи створення рекомендацій, які застосовуються для прогнозування уподобань користувачів та надання їм пропозицій, згідно їх уподобань.

Методи дослідження. Досягнення мети атестаційної роботи базується на комплексному використанні методів теорії фільтрації інформації, що стосуються створення рекомендаційних систем різних видів.

Послугами готелю користуються багато людей, особливо які хочуть гарно відпочити. Перед адміністрацією готелю постає ряд задач: вести облік клієнтів та номерів, фінансів, формування звітів для бухгалтерії, власників і податкової служби, контроль доходів. Більшість готелів нашого часу використовують різноманітні програми, в які і записується вся необхідна інформація, та є такі готелі, які ведуть журнали. Коли діло доходить до пошуку клієнта, який вже є в журналі, але був внесений декілька місяців назад, то адміністратор витрачає доволі багато часу, щоб його знайти. Ще один недолік в тому, що потрібно вручну перераховувати весь дохід та кінцеву цінність, витрати і робити звіти, переписуючи ці дані. Для того, щоб автоматизувати всі ці рутинні справи, була розроблена системи автоматизації бронювання місць в готелі.

Основна мета рекомендаційної системи автоматизації бронювання місць

в готелі – дати адміністрації готелю всі необхідні функції, які б дозволили автоматизувати процес пошуку клієнту вільного номера, або пошуку клієнта, занесення в базу даних всієї необхідної інформації про клієнта і номер, а також створення заявок на бронювання. Дана серверна частина програми дозволить позбавитися від незручностей, які виникають при пошуку, додаванню і формуванню необхідної інформації.

Адміністратору готелю доступний широкий набір функцій, які дозволяють майже повністю автоматизувати процес керування готелем.

Для створення серверної частини прикладної програми були використані наступні засоби: ОС Microsoft Windows 10, IBM RationalRose 7.0, Microsoft Word 2016, MySQLWorkbench 6.3, C# 5.0, Microsoft Visual Studio 2017.

1. ОГЛЯД І АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ, ЩО РОЗГЛЯДАЄТЬСЯ

1.1 Визначення рекомендаційної системи

Рекомендаційні системи – представляють собою підклас систем фільтрації інформації, це програми та сервіси, що намагаються передбачити які об'єкти (фільми, музика, новини, товари, веб-сайти) будуть цікаві користувачеві, спираючись на деяку інформацію (профіль користувача чи опис об'єкту). Ці програми вдосконалили способи взаємодії між сервісом та відвідувачем, тому що замість того, щоб надавати статичну інформацію, користувач отримує інтерактивні можливості [4].

Рекомендаційні системи використовуються у найрізноманітніших областях і найчастіше вважаються генераторами списків відтворення для відео- і музичних сервісів, таких як Netflix, YouTube і Spotify, рекоменаторами продуктів для сервісів, таких як Amazon, або рекоменаторами контенту для платформ соціальних мереж, таких як Facebook і Twitter [5, 6]. Найчастіше такі системи використовуються у комерційних сервісах. Рекомендаційні системи є корисною альтернативою пошуковим алгоритмам, оскільки вони допомагають користувачам виявляти елементи, які в іншому випадку, вони могли і не знайти. Для інтернет-магазинів це в принципі надзвичайно важлива функція та один з небагатьох способів якісно працювати. Рекомендації в даному випадку не є звичайною додатковою опцією платформи, вони забезпечують зручність навігації користувача по веб-ресурсу. Якщо електронний каталог містить більше 20 000 найменувань продукції, орієнтація вже представляється непомірно важкою, що ж говорити, якщо товарів мільйони.

Існує безліч конкретних прикладів використання рекомендаційних

систем найрізноманітніших видів. Рекомендації на відомому онлайн-гіпермаркеті Amazon виглядають наступним чином (рисунок 1.1).



Рисунок 1.1 – Рекомендації товарів в онлайн-гіпермаркеті Amazon

Як показує практика, не тільки онлайн-магазини використовують подібний прийом. Соціальні мережі також не відстають. На рисунку 1.2 наведено приклад із соціальної мережі Instagram.

Також подібні прийоми легко можна побачити на різних соціальних платформах, порталах, присвячених літературі, подорожам, на новинних ресурсах, словом - майже всюди. Ця методика дійсно дуже популярна. Веб-ресурс «Кинопоиск» – ще один доступний приклад (рисунок 1.3):

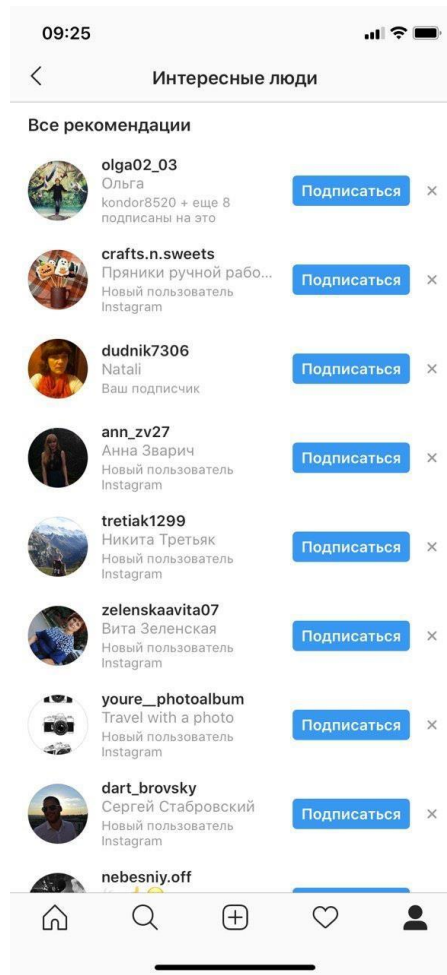


Рисунок 1.2 – Рекомендації у соціальній мережі Instagram



Рисунок 1.3 – Рекомендації у веб-каталозі присвяченому фільмам «Кинопоиск»

На разі існують три основні та найбільш використовувані стратегії по створенню рекомендаційних систем [1 – 3]:

- content-based filtering (фільтрація на основі вмісту / контентна фільтрація);
- collaborative filtering (спільна фільтрація);
- гібридний підхід – використовує обидві основні стратегії у певній мірі.

Далі будуть детально розглянуті принципи роботи цих стратегій та проаналізована можливість вирішення поставленого завдання дослідження за їх допомогою.

1.2 Рекомендаційні системи на основі спільної фільтрації

Спільна фільтрація – це триступеневий процес, що починається зі збору користувацької інформації. Потім будується матриця для розрахунку асоціацій і, нарешті, дається дуже вірогідна рекомендація. Її основне припущення полягає в наступному: ті, хто однаково оцінював будь-які предмети в минулому, схильні давати схожі оцінки інших предметів і в майбутньому [7]. Наприклад, за допомогою колаборативної фільтрації музичний додаток здатен прогнозувати, яка музика сподобається користувачу, маючи неповний список його переваг (симпатій і антипатій). Прогнози складаються індивідуально для кожного користувача, хоча використовувана інформація зібрана від багатьох учасників.

Спільна фільтрація поділяється на два основних підходи:

- memory based (заснований на сусідстві) – існує два основні різновиди цього методу: на основі користувачів, що займаються пошуком, і на основі елементів, що утворюють ту чи іншу категорію;
- model based (заснований на моделі) – даний підхід надає рекомендації, вимірюючи параметри статистичних моделей для оцінок

користувачів, побудованих за допомогою таких методів як, метод байесовських мереж, кластеризації, латентної семантичної моделі, такі як сингулярне розкладання, імовірнісний латентний семантичний аналіз, прихований розподіл Діріхле і марківський процес прийняття рішень на основі моделей. Моделі розробляються з використанням інтелектуального аналізу даних, алгоритмів машинного навчання, щоб знайти закономірності на основі навчальних даних. Цей підхід є більш комплексним і дає більш точні прогнози, так як допомагає розкрити латентні фактори, що пояснюють спостережувані оцінки. Даний підхід має ряд переваг. Він обробляє розріджені матриці краще, ніж підхід заснований на сусідстві, що в свою чергу допомагає з масштабністю великих наборів даних[7].

Розглянемо детальніше підхід заснований на сусідстві. Існує два основні методи «підбору сусідів»: «користувач-користувач», «продукт-продукт».

Для створення нових рекомендацій, для активного користувача, метод «користувач-користувач» намагається знайти користувачів із найбільш схожим «профілем взаємодій» (таких користувачів називають найближчими сусідами) для пропонування продуктів, що є найбільш популярними серед цієї групи користувачів. Тобто ідея, на якій ґрунтується такий метод – шукати користувачів, чії смаки схожі на переваги нашого цільового користувача [8]. Якщо раніше «користувач 1» і «користувач 2» проставили схожі оцінки для декількох фільмів, то ми вважаємо, що смаки у них подібні, і за рейтингами тих чи інших фільмів, проставленим «користувачем 1», можемо вгадати невідомі рейтинги «користувача 2».

Наприклад, якщо Жану - П'єру сподобалися фільми «Повернення джедая» і «Імперія завдає удару у відповідь», а Джейсону сподобався фільм «Повернення джедая», то ми безумовно повинні підказати Джейсону ще й фільм «Імперія завдає удару у відповідь» (рисунок 1.4).

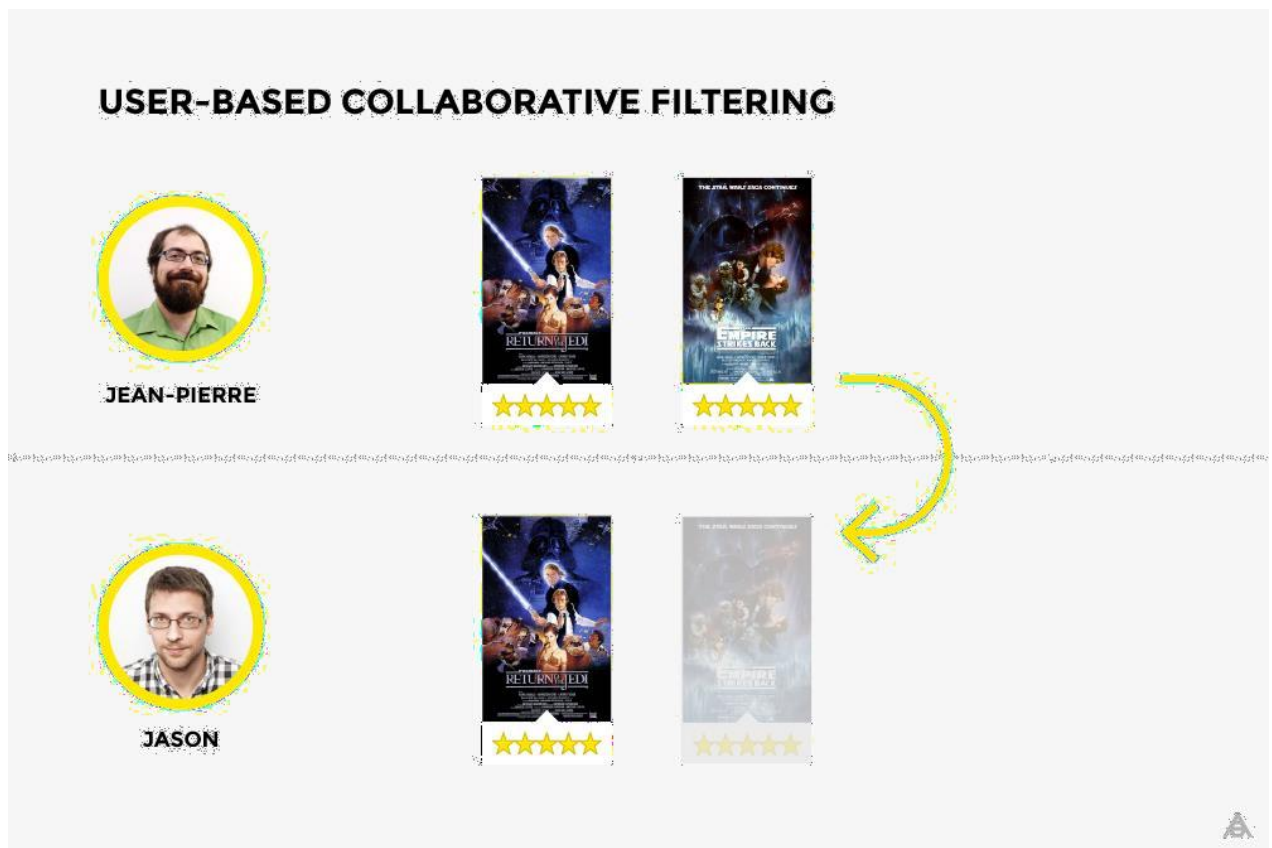


Рисунок 1.4 – Колаборативна фільтрація «користувач-користувач»

Уподобання користувачів записуються на матриці «користувачі-елементи» (рисунок 1.5), на основі якої і знаходять найближчих сусідів.

У такій матриці рядки відповідають користувачам, а стовпці – елементам. Тобто кожен рядок матриці становить вектор користувацького уподобання стосовно представлених елементів (при цьому якщо користувач, ще не оцінив той чи інший елемент, відповідний елемент матриці залишається пустим). Далі обчислюється міра схожості (за однією з обраних метрик) між активним користувачем та іншими користувачами, яка полягає у наступному: два користувача зі схожими оцінками на одному і тому ж наборі елементів мають визнаватися схожими. На основі розрахованих мір схожості формуються групи користувачів яким пропонуються елементи найбільш популярні серед них, обираючи лише ті, які активний користувач ще не оцінив.

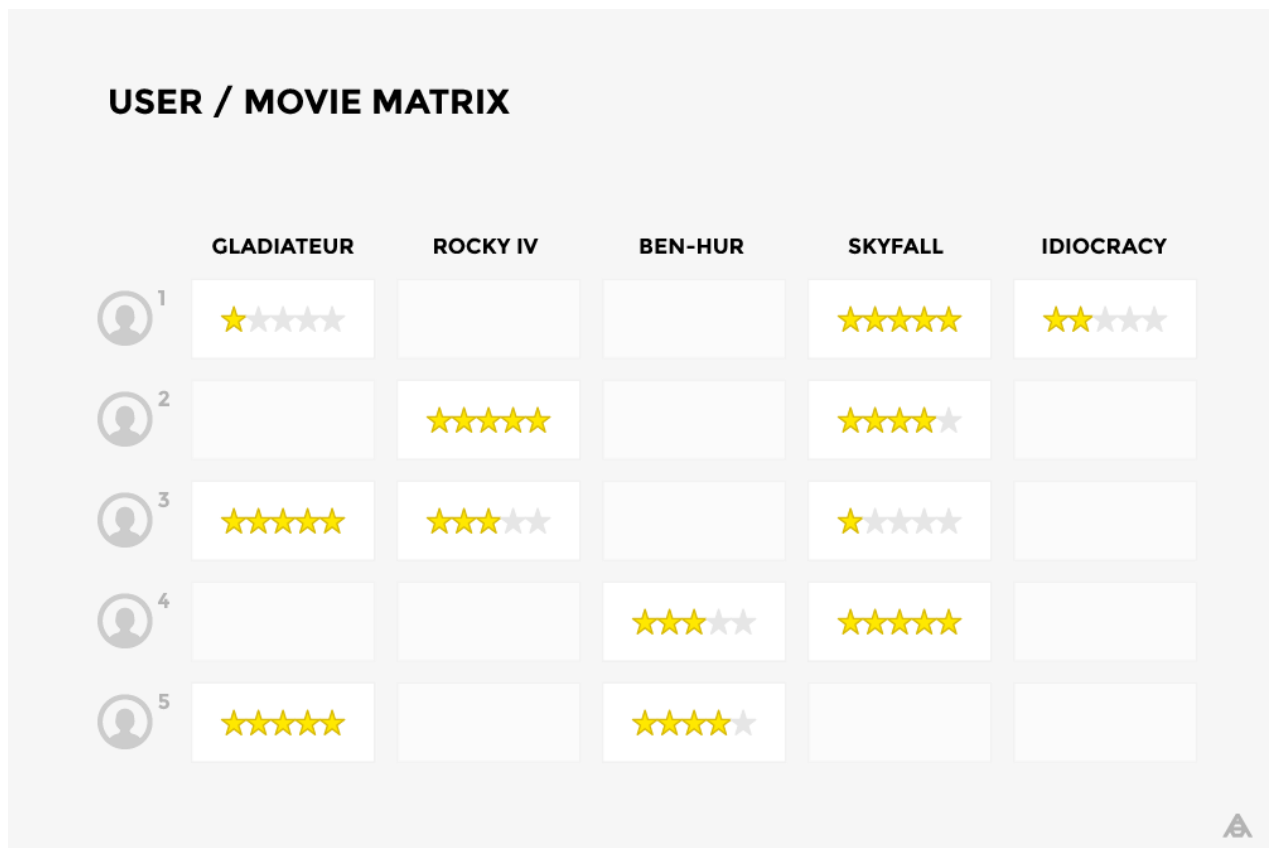


Рисунок 1.5 – Матриця оцінок «користувач-елемент»

Метод «продукт-продукт» полягає у тому, що користувачу потрібно рекомендувати елементи, схожі на ті, які він уподобав раніше. Тобто схожість розраховується між елементами, а не користувачами. При цьому, два елементи вважаються схожими, якщо більшість користувачів взаємодіяли з обома схожим чином[9]. Наприклад, Жану-П'єру і Джейсону сподобалися фільми «Повернення джедая» і «Імперія завдає удару у відповідь». Можна зробити висновок, що більшості користувачів, які високо оцінили перший фільм, повинен сподобатися і другий. Таким чином, було б доречно запропонувати фільм «Імперія завдає удару у відповідь» Ларрі, якому теж сподобався фільм «Повернення джедая» (рисунок 1.6).

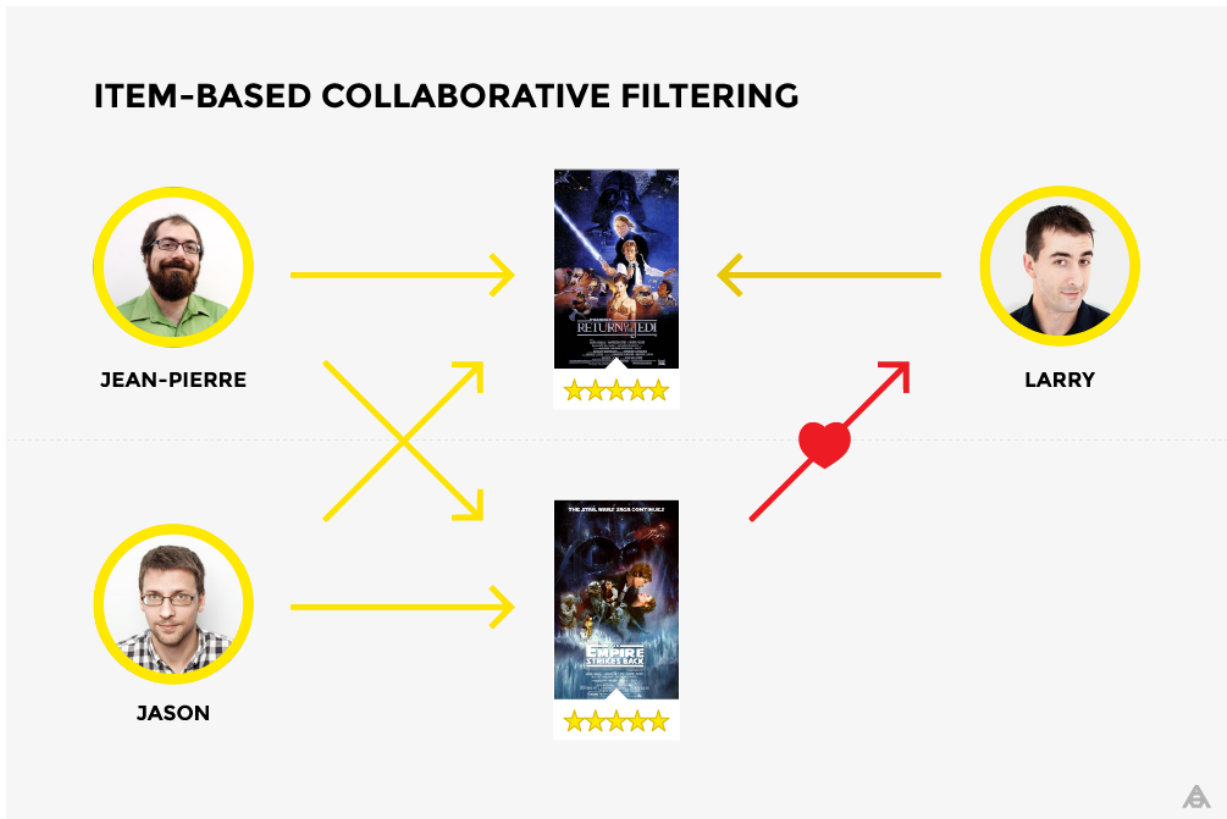


Рисунок 1.6 – Колаборативна фільтрація «елемент-елемент»

Цей підхід також використовує матрицю представлену на рисунку 1.5. При цьому розглядаються стовпці, які представляють собою вектор взаємодії елемента з кожним користувачем. Для надання рекомендації активному користувачу, обираються елементи, які він найбільше уподобав. Вони представляються, як вектори взаємодій із кожним користувачем. Далі обчислюється міра схожості між цими елементами та усіма іншими та обираються найбільш схожі, які з'являться потім у рекомендаціях.

Метод «користувач-користувач» базується на пошуку схожих користувачів на основі взаємодій з продуктами. Якщо кожен користувач взаємодіє з невеликою кількістю продуктів, метод стає достатньо чутливим до будь-яких збережених взаємодій. З іншого боку, якщо фінальна рекомендація базується лише на збережених взаємодіях користувачів із схожими інтересами дозволяє отримати більш персональний результат (низька похибка).

Метод «продукт-продукт», навпаки, базується на пошуку схожих продуктів на основі взаємодій користувачів з ними. Загалом, якщо багато користувачів взаємодіяли з продуктом, пошук сусідів є значно менш чутливим до одиничних взаємодій. Аналогічно, дії будь-яких користувачів (навіть тих, що достатньо відрізняються від обраного) розглядаються у рекомендації, роблячи цей метод менш персональним (більша похибка). Таким чином, цей підхід є менш персональним ніж підхід «користувач-користувач», але є більш надійним.

В ході аналізу підходів спільної фільтрації був виявлений ряд її ключових недоліків:

- розрідженість даних – більшість комерційних рекомендаційних систем заснована на великій кількості даних (товарів), в той час як більшість користувачів не ставить оцінки товарам. В результаті цього матриця «предмет-користувач» виходить дуже великою і розрідженою, що викликає особливі проблеми при обчисленні рекомендацій;

- масштабованість – зі збільшенням кількості користувачів в системі з'являється проблема масштабованості. Наприклад, маючи 10 мільйонів покупців і мільйон предметів складність алгоритму колаборативної фільтрації є дуже високою, а сам алгоритм вже занадто складний для розрахунків;

- холодний старт – нові предмети або користувачі представляють велику проблему для рекомендаційних систем. Такі предмети, ще не мають оцінок користувачів, тому вони не будуть потрапляти до рекомендацій поки матриця «предмет-користувач» не буде наповнена певною кількістю значень. Однак проблема з наданням рекомендації для нового користувача полягає у тому ж: користувач ще не має матриці переваг, а тому його не можливо віднести до жодної з груп [10,11].

Враховуючи суть, переваги та недоліки описаної стратегії, для вирішення поставленого завдання по створенню рекомендаційної системи для

сервісів зі «складною», з точки зору надання рекомендацій, сферою застосування, вона не може бути застосована з наступних причин:

- активність користувачів на таких сервісах буде тимчасовою, а матриці «елемент-користувач» будуть дуже розрідженими, адже користувач буде оцінювати лише певну обмежену групу товарів (наприклад двокімнатну квартиру у певному районі з обмеженою максимальною вартістю оренди). Тому проблема розрідженості даних, що супроводжує сумісну фільтрацію стає більш загальною і унеможливорює використання цього методу;

- специфічність обраної предметної сфери не дозволяє створювати навіть відносно постійні профілі користувачів. Існує дуже мала імовірність того, що користувач буде декілька разів шукати дуже схожий за параметрами елемент, а спрогнозувати його наступну активність неможливо. Отже основне припущення, що допускає спільна фільтрація у рамках такої системи недійсне для кожного конкретного користувача. Тобто групи користувачів, що формуються, будуть непостійними, і оцінки такої групи буде складно використовувати для прогнозування переваг нового користувача;

- проблема «холодного старту» не може існувати у рамках задачі, що розглядається, адже тут спостерігається постійна «текучість» активних елементів.

1.3 Рекомендаційні системи на основі вмісту

Методи фільтрації на основі вмісту засновані на описі елемента та профілі переваг користувача. Ці методи найкраще підходять для ситуацій, коли відомі дані про об'єкт (ім'я, місце розташування, опис тощо.), але не про користувача. Контент-орієнтовані рекомендації розглядають рекомендації як специфічну для користувача проблему класифікації і створюють класифікатор для симпатій і антипатій користувача на основі характеристик продукту.

У цій системі ключові слова використовуються для опису елементів, а

профіль користувача створюється для визначення типу елемента, який подобається цьому користувачеві. Іншими словами, ці алгоритми намагаються рекомендувати елементи, які схожі на ті, які користувач любив в минулому або вивчає в даний час. Зокрема, різні елементи-кандидати порівнюються з елементами, раніше оціненими користувачем, і рекомендуються найбільш підходящі елементи. Цей підхід має своє коріння в пошуку інформації та дослідженнях інформаційної фільтрації.

Ключова проблема з фільтрацією на основі контенту полягає в тому, чи здатна система дізнатися користувальницькі переваги з дій користувачів щодо одного джерела контенту і використовувати їх в інших типах контенту. Наприклад, рекомендація новинних статей на основі перегляду новин корисна, але буде набагато корисніше, коли на основі перегляду новин можна рекомендувати музику, відео, продукти тощо. Для подолання цієї проблеми, більшість заснованих на контенті рекомендаційних систем використовують гібридні системи, що містять у собі елементи обох підходів.

Даний підхід використовує опис або атрибути елементів, з якими взаємодіяв користувач, для рекомендації схожих елементів. В такому випадку залежність створюється тільки на основі попереднього вибору користувача, що робить цей метод надійним, щоб уникнути проблеми холодного запуску[2].

Припустимо, деякий користувач дивиться фільм певного жанру, йому будуть рекомендуватися фільми з відповідним жанром. Назва, рік випуску, режисер, акторський склад, іншими словами контент, опис фільму, також допомагають ідентифікувати схожий за контентом фільм. Візуалізацію цього підходу можна побачити на рисунку 1.7.

При такому підході зміст продукту вже оцінюється на основі уподобань користувача (профіль користувача), а значення атрибутів елемента - це неявні функції, які будуть використовуватися для створення профілю елемента. Потім оцінка предмета прогнозується з використанням обох профілів, і

можуть бути зроблені рекомендації[2].

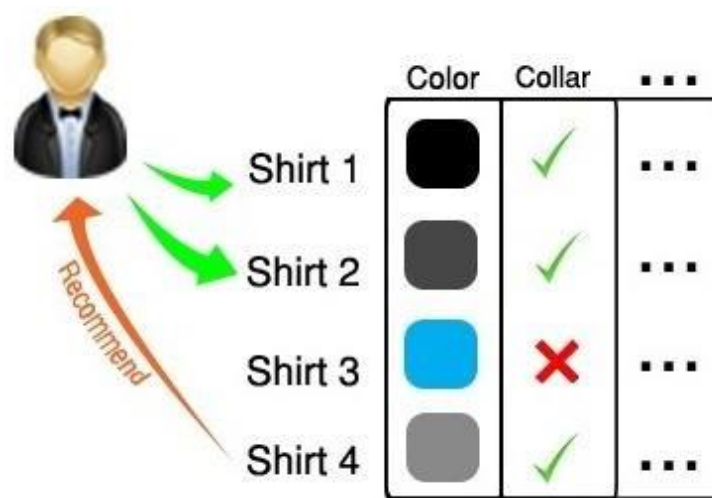


Рисунок 1.7 – Візуалізація підходу зі створенням профілю користувача і профілю елемента

Для виконання результуючої рекомендації необхідно створити профілі користувача та елементів. Для цього є дві групи вхідних даних:

- таблиця уподобань користувача (rating table): демонструє відношення користувач-елемент;
- профіль користувача (item profile): демонструє відношення елемент-атрибут, набір значень атрибутів елемента.

Тоді на основі цих даних створюється профіль користувача (user profile), який демонструє відношення користувач-атрибут, тобто маємо набір значень атрибутів елемента деякого типу, яким найбільше віддає перевагу користувач[3].

Тепер, маючи профілі користувача та елементів, можна передбачити оцінку для елементів, відносно вподобань даного користувача.

Даний підхід має ряд переваг відносно спільної фільтрації:

- незалежність користувача: для спільної фільтрації потрібна оцінка інших користувачів, щоб знайти схожість між користувачами, а потім дати рекомендацію. Замість цього метод, заснований на контенті, повинен тільки

проаналізувати елементи і профіль користувача для рекомендації;

- прозорість: метод спільної фільтрації дає рекомендацію, бо деякі невідомі користувачі мають такий же смак, як ви, але метод, заснований на контенті, рекомендує елементи тільки на основі їх опису;

- відсутність холодного старту: на відміну від спільної фільтрації, можна пропонувати нові елементи, перш ніж вони будуть оцінені значним числом користувачів.

Також цей підхід має ряд суттєвих недоліків:

- обмежений аналіз контенту: якщо контент не містить достатньо інформації для точного розрізнення елементів, рекомендація буде не зовсім остаточною;

- над спеціалізація: метод заснований на контенті забезпечує граничний ступінь новизни, оскільки він повинен відповідати особливостям профілю та елементів. Цілком досконала фільтрація на основі вмісту може не запропонувати нічого нового;

- новий користувач: коли недостатньо інформації для створення надійного профілю користувача, рекомендацію неможливо надати правильно. Незважаючи на переваги даного методу стосовно спільної фільтрації він також не придатний для вирішення поставленої задачі адже для його коректної роботи як і для методів спільної фільтрації необхідна можливість створення надійного користувацького профілю на основі рейтингів, що у рамках об'єкту дослідження є неможливим.

1.4 Рекомендаційні системи засновані на знаннях

Системи рекомендацій, засновані на знаннях (експертні рекомендаційні системи), являють собою особливий тип систем рекомендацій, які засновані на явних знаннях про асортимент товарів, користувацьких перевагах і критеріях

рекомендацій (тобто, який товар слід рекомендувати, в якому контексті) [12].

Ці системи застосовуються в сценаріях, де основні підходи, такі як спільна фільтрація і фільтрація на основі контенту, не можуть бути застосовані. Системи рекомендацій, засновані на знаннях, добре підходять для складних областей, де предмети купуються не дуже часто, таких як квартири і автомобілі [13]. Іншими прикладами предметних областей, релевантних для систем рекомендацій, заснованих на знаннях, є фінансові послуги і туризм. Системи, засновані на рейтингах, не працюють добре в цих областях через малу кількість доступних рейтингів. Крім того, в доменах зі складними предметами, клієнти хочуть явно вказати свої переваги (наприклад, «максимальна ціна автомобіля-Х»). У цьому контексті система рекомендацій повинна враховувати обмеження: наприклад, слід рекомендувати тільки ті фінансові послуги, які підтримують інвестиційний період, зазначений клієнтом. Жоден з цих аспектів не підтримується такими підходами, як спільна фільтрація і фільтрація на основі контенту. Ще однією перевагою, заснованих на знаннях, рекомендаційних систем є відсутність проблем з холодним запуском.

Існує декілька основних ключових ідей які використовуються для створення таких рекомендаційних систем:

- «діалог» – системи рекомендацій, засновані на знаннях, часто є діалоговими, тобто користувацькі переваги виявляються в рамках циклу зворотного зв'язку [14]. Основною причиною діалогового характеру цих систем є складність предметної області, в якій часто неможливо сформулювати всі вимоги користувача одночасно. Крім того, уподобання невідомі точно на початку, але можуть бути виявлені в рамках сеансу рекомендацій;

- «обмеження» – зворотний зв'язок з користувачем може бути реалізований у вигляді відповідей на питання, які обмежують набір відповідних елементів[15]. Прикладом такого питання є «Якому типу системи об'єктивів ви

віддаєте перевагу: фіксовані або змінні об'єктиви?». На технічному рівні такі сценарії рекомендацій можуть бути реалізовані на основі систем рекомендацій, заснованих на обмеженнях що встановлюються користувачем під час виконання пошукового запиту (пошуковий фільтр). В такому разі рекомендаційна система є надбудою над обмеженнями, що встановив користувач, яка на основі встановлених асоціативних правил робить пошук відповідних варіантів. Необхідність у такій надбудові є коли простір елементів є складним і його звичайна фільтрація (пошуковий фільтр) призведе до втрати деяких варіантів;

– «критика» – користувацький зворотний зв'язок зазвичай надається в термінах «критичних зауважень», які вказують запити на зміну, що стосуються елемента, в даний час рекомендованого користувачеві. [16] Критика потім використовується для рекомендації наступного «кандидата». Приклад критики в контексті сценарію рекомендації цифрової камери: «Я хотів би мати таку камеру, але з більш низькою ціною». Це приклад «критичної оцінки», яка представляє запит на зміну одного атрибута елемента.

Найбільшим недоліком такого підходу є вузьке місце набуття знань, викликане необхідністю явного визначення рекомендацій [12]. Адже для створення такої системи рекомендацій необхідна участь експертів предметної галузі, які побудують систему асоціативних правил, на яких будуть базуватися рекомендації. Такий спосіб знань є найбільш затратним з точки зору фінансів, людських ресурсів та часу. До того ж при такому підході кожна така система рекомендацій буде унікальною і її не можливо буде застосувати до іншої предметної області без повторної побудови нових асоціативних правил.

2 РЕКОМЕНДАЦІЙНА СИСТЕМА ДЛЯ СЕРВІСІВ ЗІ «СКЛАДНИМИ» ПРЕДМЕТНИМИ СФЕРАМИ

2.1 Стратегія надання рекомендацій

Аналіз основних стратегій по створенню рекомендаційних систем показав, що вони не можуть бути застосовані на пряму для вирішення поставленої задачі через необхідність використання та зберігання історії рейтингів. А рекомендаційні системи засновані на знаннях, хоча і призначені для роботи з такими предметними сферами, не є універсальними через необхідність явного збору знань для кожної предметної області, та явній побудові асоціативних правил взаємодії з залученням експертів предметної області.

Можна зробити висновок, що існує необхідність у розробці стратегій та способів прогнозування уподобань користувачів, які могли б працювати зі складними предметними сферами, не потребуючи при цьому явного збору знань для кожної такої сфери окремо. Тобто така стратегія має відповідати наступним двом пунктам:

- незалежність від історії збережених рейтингів;
- незалежність логіки побудови прогнозів від предметної сфери.

Деякі аспекти розглянутих методів можна використати та об'єднати між собою, для створення системи рекомендацій що здатна працювати зі складними предметними областями та мінімізує вузьке місце у способі отримання знань.

По перше, можна частково застосувати метод фільтрації заснований на контенті, але в цьому разі буде аналізуватися лише опис товарів без побудови профілю користувача [17]. Стратегія полягає у наступному: користувачу пропонуються об'єкти, які максимально схожі на вподобаний ним об'єкт, та цьому «вподобаним» вважається об'єкт, що переглядається користувачем у

даний момент. Тобто в основу такої рекомендації входить так званий «метод ідеальної точки», ідеальною точкою при цьому і є об'єкт, що вподобав користувач (рисунок 2.1). Такий підхід дозволить знаходити елементи, які не зовсім відповідають пошуковому запиту (фільтру) користувача, що є першочерговою задачею.

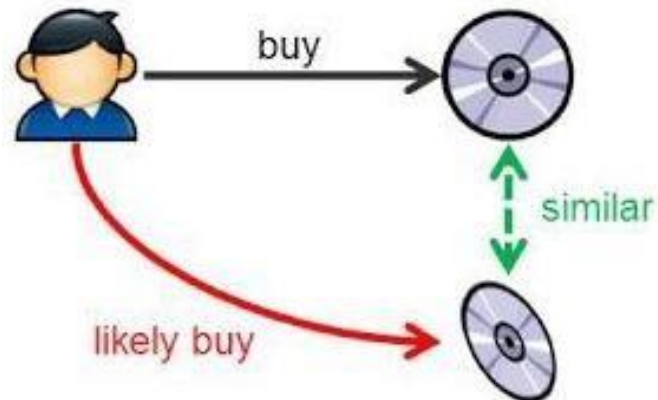


Рисунок 2.1 – Підхід з аналізом лише контенту

Для знаходження «однаковості» елементів необхідно використовувати різноманітні метрики подібності для порівняння значень атрибутів елементів. При цьому для кожного атрибута необхідно знати дві речі: його відносну важливість, та вигляд функції яка описує подібність двох значень цього атрибуту.

По друге, можна використати ідею «діалогу», яка стосується експертних рекомендаційних систем, для організації зворотного зв'язку з користувачем. Тобто уподобання користувача будуть формуватися в рамках поточного сеансу перегляду елементів, а рекомендації будуть формуватися шляхом аналізу опису тих елементів, що становлять його уподобання[18].

Тоді створення рекомендацій буде становити двоступеневий процес. На першому етапі користувачу пропонуються оголошення, що найбільш схожі на те, яке він наразі переглядає (активне оголошення). На другому етапі, коли користувач вже переглянув певну кількість оголошень, рекомендації

базуються на групі з деякої кількості останніх переглянутих оголошень.

Група останніх переглянутих оголошень формується методом «ковзаючого вікна». Це алгоритм трансформації, що дозволяє сформувати з членів часового ряду набір даних, який може служити навчальною множиною для побудови моделі прогнозування [27]. Під «вікном» у даному випадку розуміється часовий інтервал, що містить набір значень, які використовуються для формування навчального прикладу. У процесі роботи алгоритму вікно послідовно зміщується на одиницю спостереження, при цьому «забуваючи» попередні спостереження (рисунок 2.2). Такий підхід чудово реалізує ідею «діалогу», враховуючі поточні уподобання користувача та враховуючи їх зміну шляхом «забування» старих елементів.

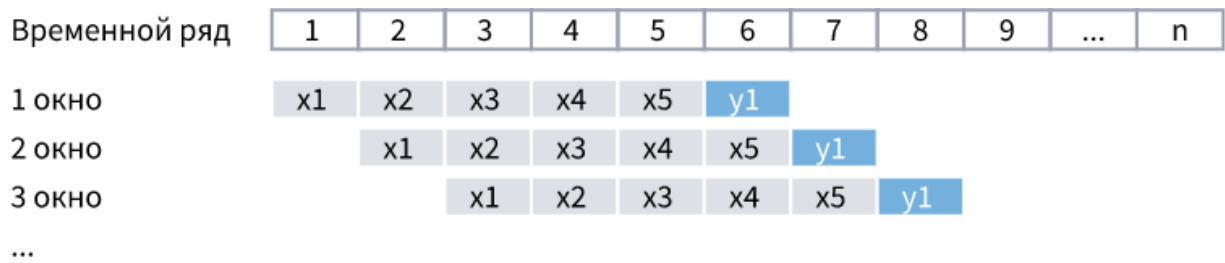


Рисунок 2.2 – Метод «ковзаючого вікна»

По-третє, можна застосувати ідею «обмежень» шляхом зменшення простору елементів, що пропонуються у якості рекомендацій, відповідно до основних обмежень у пошуковому запиті. Наприклад, не має сенсу пропонувати користувачеві квартири у Києві, якщо при пошуку він обрав Харків, навіть при ідеальній відповідності іншим обмеженням. При цьому повного обмеження відповідно до усіх вимог користувача відбуватися не повинно, щоб не створити проблему з утратою деяких елементів.

Описана стратегія має наступні ключові переваги, відносно вирішуваної проблеми:

- відсутність необхідності у створенні профілю користувача, адже система опирається лише на опис елементів;

- вирішується проблема холодного старту, адже елементи не залежать від розміру відношення «користувач-елемент»;
- уподобання користувача виявляються у реальному часі;
- присутня значна доля універсальності, адже немає необхідності для створення асоціативних правил пошуку для кожної предметної області.

Проте така стратегія буде мати і свої недоліки: оскільки подібна система рекомендацій у повній мірі опирається на опис елементів, для правильного розрізнення елементів, необхідно створити дуже детальний та змістовний опис кожного елемента, а також обов'язковою умовою є необхідність максимального заповнення інформацією профілю об'єкту.

Також такий підхід не буде універсальним у повній мірі адже експерту предметної області все ж необхідно надати такій системі рекомендацій відносну важливість атрибутів що описують елементи, та спосіб їх порівняння. Проте відносну важливість атрибутів можна встановити і без участі експерта виходячи із законів здорового глузду і принципів функціонування предметної області, або виходячи із обмежень що встановив користувач під час виконання пошукового запиту. А вигляди функцій близькості можна встановлювати вже на етапів провадження рекомендаційної системи, виходячи із набору стандартних метрик, що існують наразі для характеристик, що представлені у тих чи інших шкалах.

Також суттєвим обмеженням є те, що така система передбачає, що рекомендації мають відбуватися на множині однакових об'єктів, що характеризуються певним набором однакових характеристик. Тобто рекомендація буде працювати лише в межах обраної рубрики, рекомендуватися буде лише той самий тип елементів.

2.2 Математичний опис задачі

Запропонована стратегія передбачає, що кожен елемент

характеризується вектором значень деяких визначених характеристик.

Для кількісної оцінки схожості деякого елемента з «еталонним» використовується метод зваженої суми критеріїв [20]. Для цього кожен елемент необхідно представити вектором критеріїв:

$$K = \{k_1, k_2, k_3, k_4 \dots k_n\}, \quad (2.1)$$

де k_i – кількісна оцінка схожості i -тої характеристики елемента з i -тою характеристикою «еталонного» елемента.

Відносна важливість кожної з характеристик виражена вектором вагових коефіцієнтів:

$$W = \{w_1, w_2, w_3, w_4 \dots w_n\}, \quad (2.2)$$

де w_i – відносна важливість i -тої характеристики елемента;

$$\sum w_i = 1.$$

Результуюча оцінка схожості елемента розраховується наступним чином:

$$\text{Score} = \sum w_i \times k_i, \quad (2.3)$$

Для оцінки на основі групи елементів, оцінки (2.3) нового елемента, що розраховуються відносно кожного елемента у межах групи переглянутих, підсумовуються між собою для отримання кінцевого результату:

$$\text{Score}^* = \sum_{i=1}^N \text{Score}_i, \quad (2.4)$$

де N – кількість переглянутих елементів.

В цьому випадку подібний розрахунок можна розглядати як голосування

переглянутих сторінок за не переглянуті, та сторінка яка має найбільші значення голосів (2.3) відносно елементів у групі і отримає найвищий бал.

Оцінка близькості елементів відбувається окремо за кожною характеристикою з подальшим формуванням вектору (2.1).

Для характеристик, що представлені у інтервальній оцінці схожості характеристики елемента аналогічній характеристиці еталонного елемента буде розраховуватися за допомогою функцій затухання (або функцій розподілу).

Функції затухання оцінюють певну величину за допомогою функції, що затухає в залежності від збільшення відстані між значенням цієї величини та певним заданим джерелом (центром мас). Така оцінка схожа на запит, щодо входження деякого значення в заданий діапазон, з подальшим призначенням кількісної оцінки цьому значенню в залежності від близькості до центра діапазону.

Використовують три основні залежності (форми) розпаду (рисунок 2.3):

- гауссіан (розпад за функцією Гауса);
- експоненціальний розпад;
- лінійний розпад.

Для того щоб використати таку функцію для оцінки характеристик необхідно для кожної з них визначити значення наступних параметрів (рисунок 2.4):

- origin (центр) – значення еталонного оголошення;
- offset (зміщення) – відстань, на якій значення характеристики буде вважатися рівним еталонному значенню;
- scale (масштаб) – відстань від центру (+ зміщення) яка визначає діапазон прийнятної (допустимої) схожості;
- decay (спад) – значення схожості в яке оцінюється значення характеристики на відстані масштабу від обраного центру.

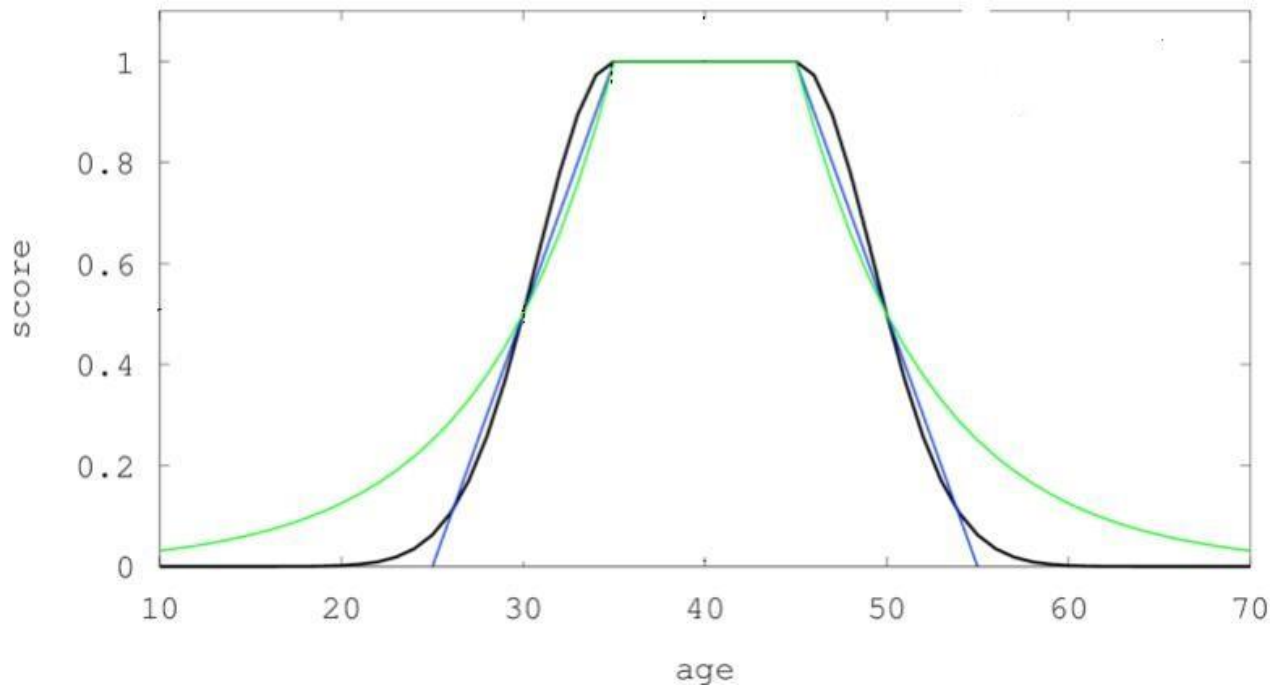


Рисунок 2.3 – графіки залежності оцінки (score) деякої величини в залежності від збільшення відстані до центра мас: зелений – експоненціальна залежність; синій – лінійна залежність; чорний – гаусіан.

Параметри Scale та Decay разом встановлюють швидкість спаду залежності як видно з рисунку 2.4.

В рекомендаційній системі, що розробляється в атестаційній роботі, використовується гаусівська залежність, оскільки лінійна залежність дуже примітивна і встановлює пропорціональне зменшення оцінки в процесі віддалення від центру, а експоненціальна залежність має дуже крутий графік залежності поряд із центром з тенденцією до більшої пологості на його кінцях, що позначатиме дуже стрімке зменшення значень оцінок в межах обраного масштабу, та більш полого – за його межами, а в даному разі необхідна зворотна залежність, яку і надає гаусіан.

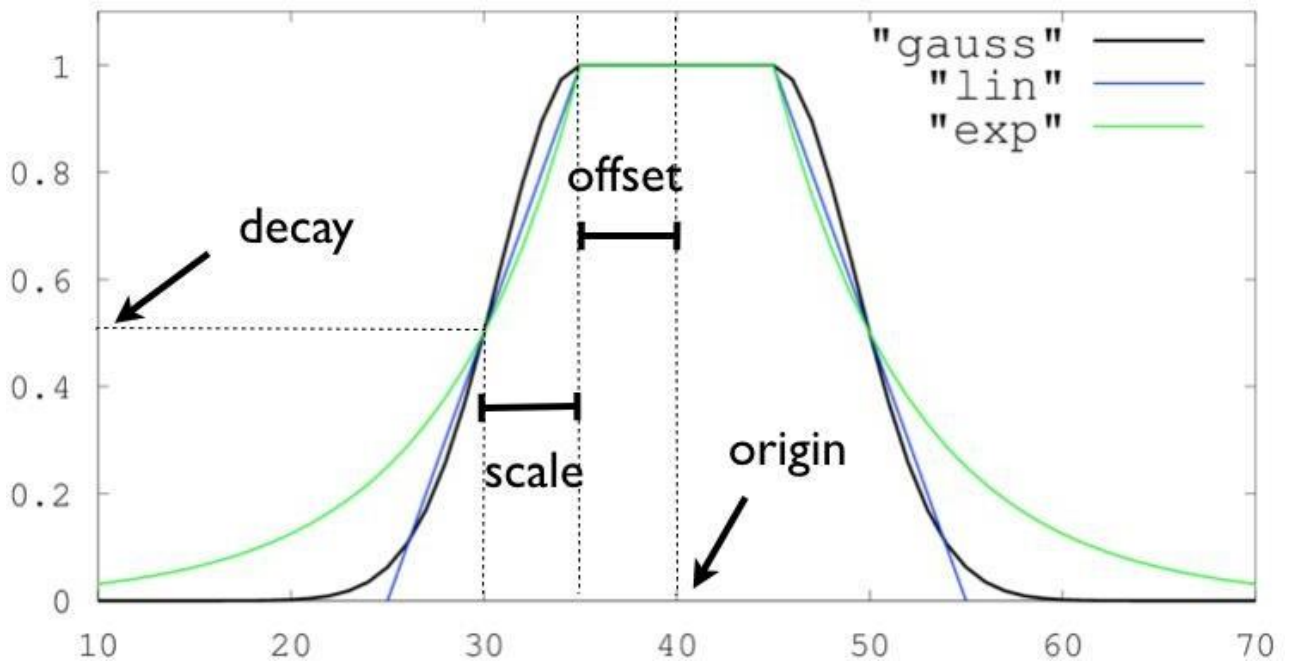


Рисунок 2.4 – Наочна інтерпретація параметрів затухання

Тоді затухання для певної характеристики деякого елемента можна визначити наступним чином:

$$K_i = \exp\left(-\frac{(|\text{value}_i - \text{origin}| - \text{offset})^2}{2 \times \left(-\frac{\text{scale}^2}{2 \times \ln(\text{decay})}\right)}\right), \quad (2.5)$$

де K_i – кількісна оцінка схожості i -тої характеристики елемента з i -тою характеристикою «еталонного» елемента, приймає значення в діапазоні $[0;1]$;

value_i – значення i -тої характеристики елемента;

origin – значення i -тої характеристики еталонного елемента;

offset – відстань, на якій значення характеристики буде вважатися рівним еталонній;

scale – відстань від центру (+ зміщення) яка визначає діапазон прийнятної (допустимої) схожості;

decay – значення схожості в яке оцінюється значення характеристики на

відстані scale від origin.

Для характеристик, що представлені у номінальних шкалах можливо встановити лише подібність або відмінність об'єктів. Тому подібність значень таких характеристик буде підраховуватися наступним чином:

- $K_i=1$, якщо значення відповідних характеристик однакові;
- $K_i= 0$, якщо значення відповідних характеристик різні.

Тобто в такому разі йдеться про повне співпадіння значення, що буде оцінюватись максимальним балом, або про відсутність співпадіння, яке оцінюється нулем.

Якщо значення деякої характеристики відсутнє, то оцінка схожості цієї характеристики елемента аналогічній характеристиці еталонного елемента буде розраховуватись наступним чином: $K_i = 0$.

В цьому випадку буде спостерігатися зменшення значення функції схожості.

3 ПОСТАНОВКА ЗАДАЧІ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

3.1 Формалізована постановка задачі

Метою даної роботи є розробка та дослідження рекомендаційних систем в різних галузях за допомогою веб-технологій. Розробити рекомендаційну систему і вирішити проблему щоб наша система працювала коректно. Розробка цієї системи потрібна для того щоб полегшити процес бронювання номерів та пошук схожих варіантів.

Більшість людей, які хочуть замовити номер, в наш час витрачають в середньому 1 годину для пошуку необхідного готелю та 1.5 години для пошуку та оформлення вибраного номеру. Через те, що не існує рекомендаційної системи, клієнт може витратити завелику кількість часу через те, що витратить багато часу на пошук готелю та бронювання номеру, також клієнт може буде не повністю задоволений вибором.

Багато клієнтів хочуть мати можливість подивитись готелі та номери, можливість швидкого оформлення і вибору підходящого для них способу оплати послуг(якщо це потрібно). Для користувачів рекомендаційна система готелю дозволить вибрати готель та номер в цьому готелі через веб-сайт, отримати знижки для постійних клієнтів, а також прискорить процес бронювання. Для адміністрації готелю система дасть можливість обліку номерів, обліку клієнтів, повної звітності по фінансам і дозволить зекономити час при реєстрації клієнтів і наступній оплаті наданих послуг. Таким чином рекомендаційна система «Готель» буде вигідна як для клієнта так і для працівників готелю, а також дозволить зекономити час обох сторін, і позбавить від появи проблем та суперечок між клієнтами і працівниками.

3.2 Вимоги до системи, що розробляється

Готель надає можливість клієнтам забронювати номер. Для того щоб продати клієнту номер, нам потрібні такі дані: прізвище, ім'я та по батькові, дата народження та стать, час приїзду та час від'їзду та суму к оплаті. Коли буде відбуватися реєстрація клієнта, він повинен бути пред'явити паспортні дані та розрахуватися за номер, якщо потрібно. Готель зберігає дані про всі номери, які зайняті клієнтами. Якщо буде якась зміна в даті бронювання номеру, усі данні повинні бути змінені в анкеті клієнта. Базою даних може користуватись як клієнт, так і адміністратор, однак їм надаються різні права доступу до неї: адміністратор може переглядати інформацію, яка йому потрібна та вносити зміну у базу даних, а клієнт може тільки переглядати інформацію. База даних повинна бути захищена логіном та паролем для входу. За допомогою бази даних ми можемо зберігати усі данні про клієнтів та номери, зберігати данні про номери які заброньовані клієнтам робити пошук потрібного номера та забезпечувати оформлення потрібного номера.

Для того щоб повністю представити функціонал сайту, виділено основні бізнес-процеси:

- а) реєстрація і авторизація в системі клієнта через веб-сайт;
- б) перегляд та бронювання номерів;
- в) створення, перегляд, змінення, фільтрація списку клієнтів готелю;
- г) авторизація в системі співробітника через прикладну програму;

3.3 Характеристика предметної області

Задачу створення рекомендаційної системи для областей, що охоплюють «складні» предметні сфери будемо виконувати на прикладі сфери готелю.

В даний час існує безліч систем, які здійснюють посередницькі послуги між клієнтом і готелем, що надає можливість забронювати номер. Як правило,

вони надають такий набір послуг:

- бронювання номерів;
- варіанти дозвілля;
- надання послуг оренди автомобіля.

Такі сервіси зазвичай складаються з клієнтського додатку, що представлений веб - інтерфейсом, що відповідає за отримання даних з сервера для подальшої візуалізації в зручному вигляді для користувача і відправки даних на сервер, а також сервера, який приймає запити клієнта, обробляє і зберігає інформацію. Сервіси з пошуку номерів надають своїм користувачам наступні загальні можливості:

- перегляд та пошук номерів за відповідними параметрами;
- фільтрація оголошень за різними критеріями;
- реєстрація користувачів у системі;
- додавання/видалення номерів за відповідними рубриками;
- написання відгуків;
- надання користувачу рекомендацій.

3.4 Огляд та аналіз існуючих аналогів

Перш ніж почати розробляти рекомендаційну систему бронювання місць в готелі, потрібно переглянути та провести аналіз існуючих аналогів, щоб не робити помилок та щось використати у своєму проекті.

Було проаналізовано два ресурси: "Booking" та "Tickets" з наданням рекомендаційних систем та три веб-ресурси, які напряду надають користувачу можливість забронювати номер на їхньому сайті. Спершу було обрано готель "Надія", наступний веб-ресурс готель "Кірофф" та останній "Kharkiv-Palace", розглянемо детальніше веб ресурс "Booking".

На рисунку 3.1 зображено ресурс, який виступає посередником (агентом) між гостями, які бажають забронювати житло, і вашим об'єктом. Такий бізнес-підхід також відомий як «агентська модель». На відміну від інших сайтів з бронювання об'єктів розміщення, компанія Booking.com не бере участь в угоді між вашим об'єктом розміщення і гостем. Це надає можливість самостійно управляти тарифами і наявністю місць, а також надавати прозору інформацію про ваші правила та пропозиції в цілому. Існує два варіанти оплати гостем вартості його бронювання: вже при виїзді, або завчасно. Якщо оплата надходить до того, як гість заселяється в об'єкт розміщення, то в разі необхідності відмінити бронь йому потрібно завчасно, або гроші не повернуться. Ви укладаєте з Booking.com угоду про використання онлайн - платформи з бронювання. Для цього потрібно буде сплатити Booking.com фіксовану суму комісії за всі реалізовані бронювання, а також бронювання по безповоротному тарифу і з частковим поверненням вартості.

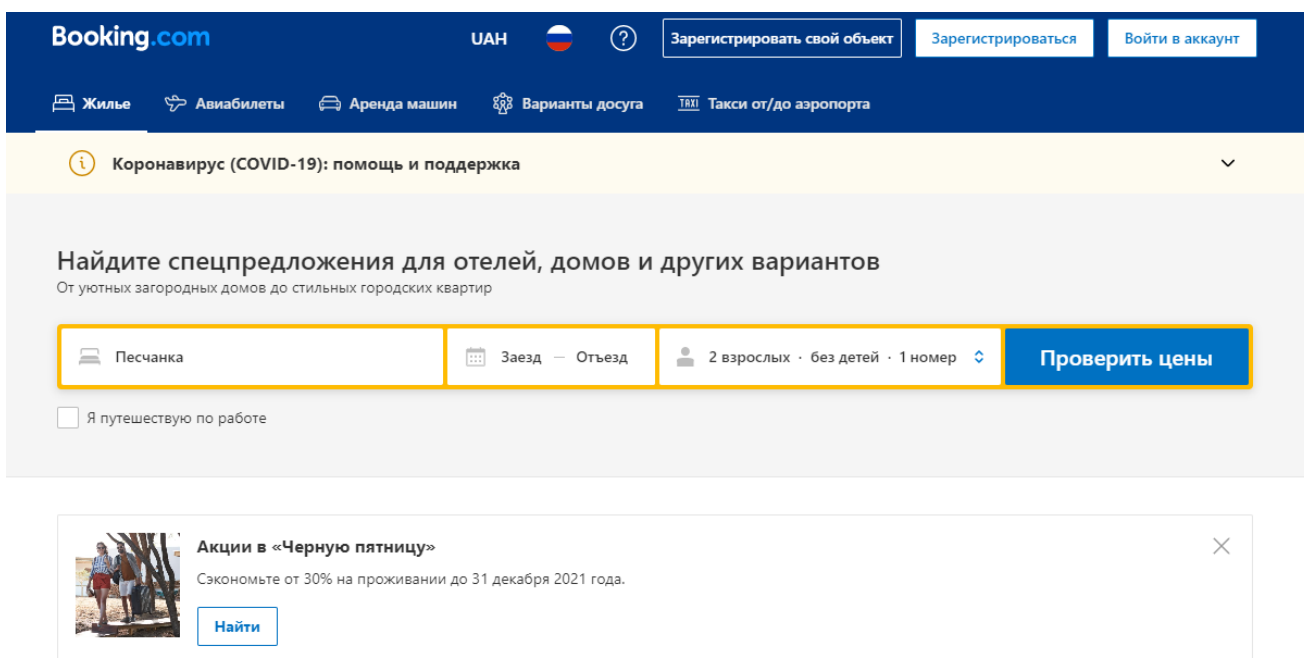


Рисунок 3.1 - Головна сторінка веб-ресурсу "Booking"

На рисунку 3.2 представлена сторінка пошуку вільних номерів. Зліва розташована форма фільтра, де користувач має вибрати назву об'єкта або місто проживання, також він має вибрати дату заїзду, дату від'їзду та вирішити на скільки чоловік вам потрібен номер. Далі рекомендаційна система вам підбере варіанти за вашими побажаннями.

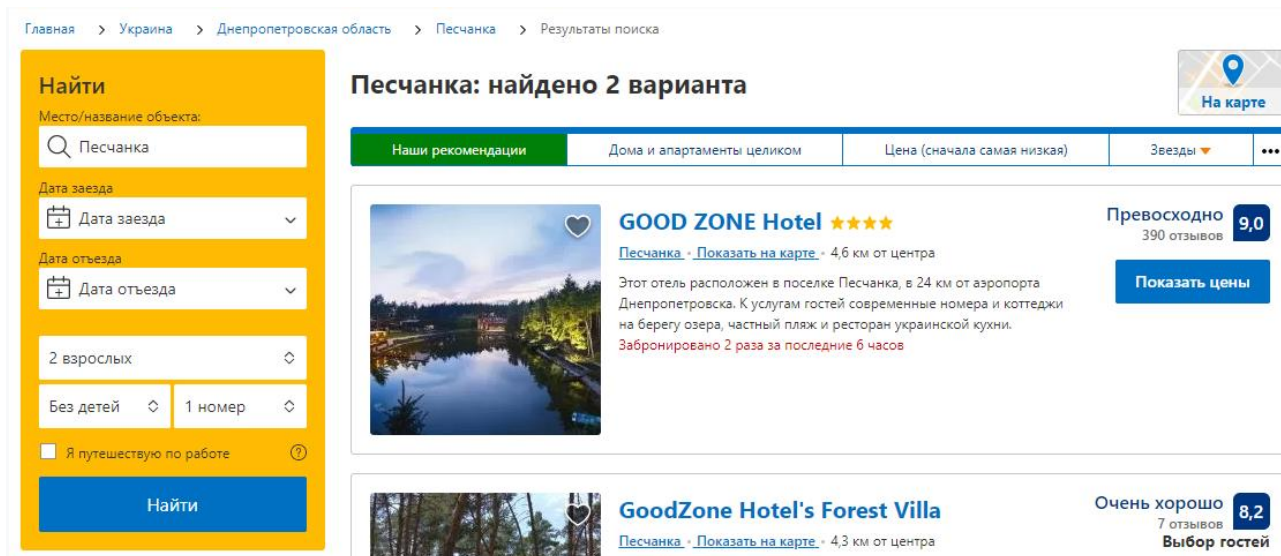


Рисунок 3.2 - Сторінка підбору/пошуку ресурсу "Booking"

На рисунку 3.5 представлена сторінка пошуку вільних номерів. Зверху розташована форма фільтра, де користувач має вибрати назву об'єкта або місто проживання, також він має вибрати дату заїзду, дату від'їзду та вирішити на скільки чоловік вам потрібен номер. Далі рекомендаційна система вам підбере варіанти за вашими побажаннями.

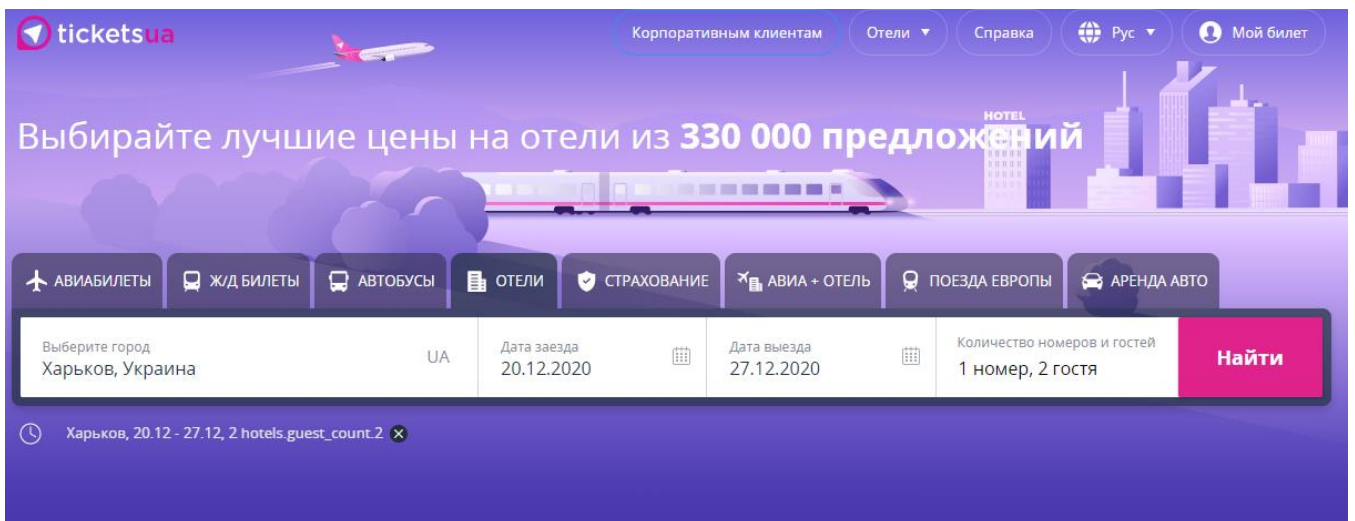


Рисунок 3.4 - Головна сторінка веб-ресурсу "Tickets "

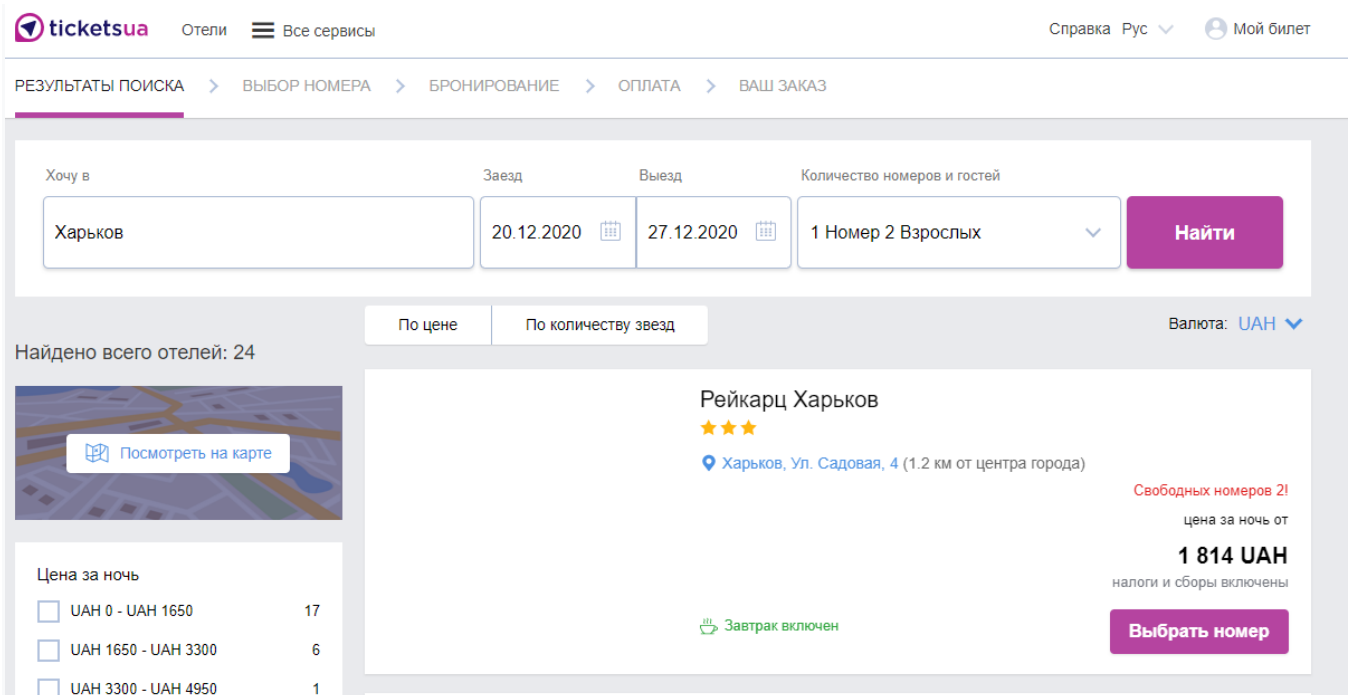


Рисунок 3.5 - Сторінка підбору/пошуку ресурсу Tickets "

Далі розглянемо додатки, які напряму надають клієнтам номери, без посередницьких послуг.

На рисунку 3.6 зображено головний вигляд сторінки цього готелю. На цій сторінці звичайний інтерфейс, який буде зрозумілий усім користувачам. На

цьому сайті є уся інформація про номери та інформація про всі послуги, які вони пропонують. Щоб переглянути інформацію про номери, потрібно натиснути кнопку "Гостиница" та ви побачите усі номери різних категорій та ціни на них.

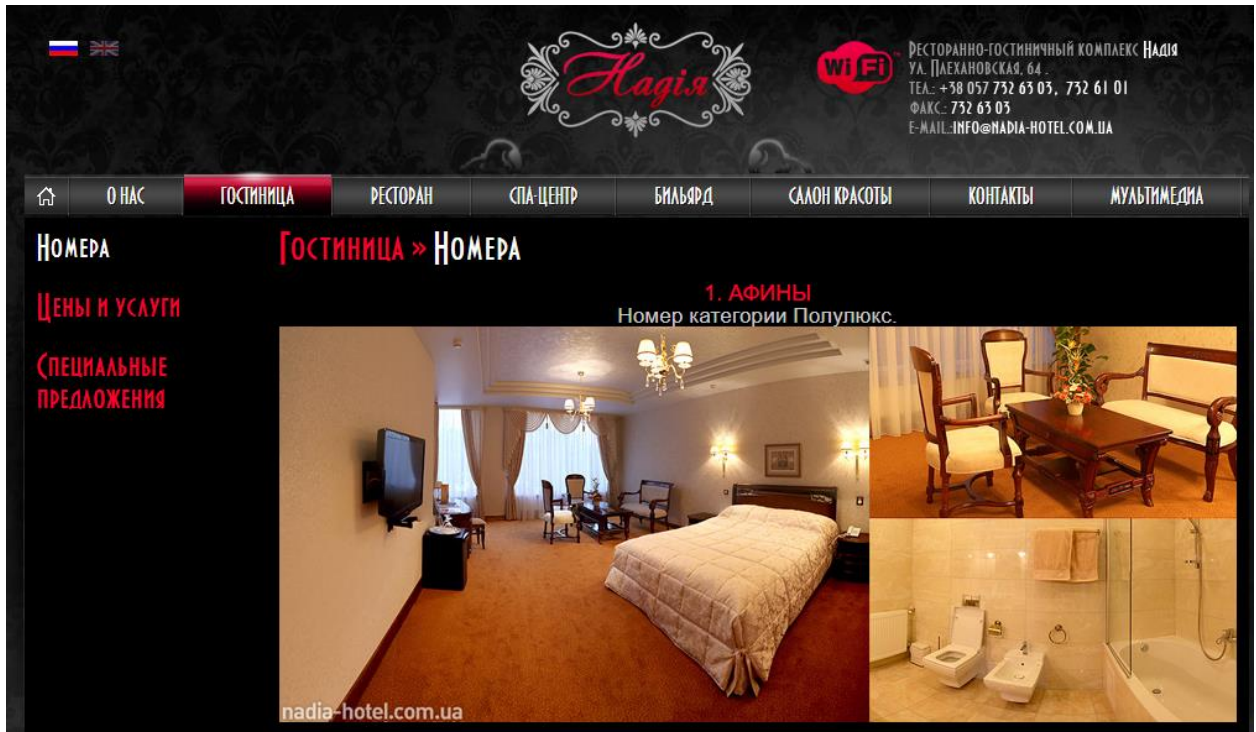


Рисунок 3.6 - Головна сторінка веб-ресурсу "Надія"

Навігація усього сайту представлена у виді верхнього меню, можна переглянути інформацію про готель, ресторан, SPA та спеціальні пропозиції. На сайті дуже гарно підібрані кольори та також при відкритті головної сторінки одразу з'являється форма з бронюванням номеру. Щоб забронювати номер у цьому готелі потрібно заповнити дату заїзду та виїзду, кількість дітей та їх вік, кількість дорослих та якщо є промокод, то ввести його.

Наступним розглянемо веб-ресурс "Кірофф".



Рисунок 3.7 - Головна сторінка готелю "Кірофф"

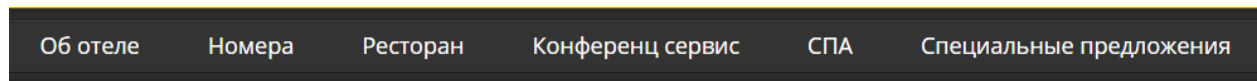


Рисунок 3.8 - Головне меню готелю "Кірофф"

Після того як заповнили поле та натиснули кнопку забронювати, веб-ресурс запропонує нам усі вільні номери.

Останнім розглянемо портал "Kharkiv-Palace".

Як і в минулих аналогах навігація представлена в вигляді горизонтального меню, можна переглянути інформацію про номери та ресторан, спа та фітнес. Над меню представлена адреса та номер мобільного телефону. Щоб забронювати номер потрібно натиснути кнопку забронювати у горизонтальному меню.

4.6 Превосходно
Отзывы гостей (45)

Booking
От : 2050 UAH

07.06.2019 - 09.06.2019 / Номеров : 1, Взрослые: 2

Выбор по номеру Выбор по тарифам Валюта отеля

<p><i>Room Only</i></p> <p>Правила и условия</p>	<p>от 2050 UAH / ночь</p> <p>ВЫБРАТЬ НОМЕР</p>
<p><i>Стандартный тариф</i></p> <p>Туристический сбор 1% не включен в стоимость При размещении детей до ...</p> <p>Правила и условия</p>	<p>от 2250 UAH / ночь</p> <p>ВЫБРАТЬ НОМЕР</p>

Рисунок 3.9 - Номери готеля "Кірофф"

ГОТЕЛЬ ХАРКІВ ПАЛАЦ

пр. Незалежності, 2
Харків 61058, Україна

+38 057 766 44 00

UA

НОМЕРИ • КОНФЕРЕНЦІЇ ТА УРОЧИСТОСТІ • ВЕСІЛЛЯ • РЕСТОРАНИ ТА БАРИ • SPA ТА ФІТНЕС • БІЛЬШЕ... • ЗАБРОНЮВАТИ

Ласкаво просимо до Kharkiv Palace!

Готель Kharkiv Palace - перший та єдиний розкішний готель найвищої категорії, що розташований в самому серці Харкова.

переглянути повне відео

Рисунок 3.10 - Головна сторінка сайту "Kharkiv-Palace"

Після того як заповнили усі данні для бронювання номеру, цей портал видає нам усі доступні номери.

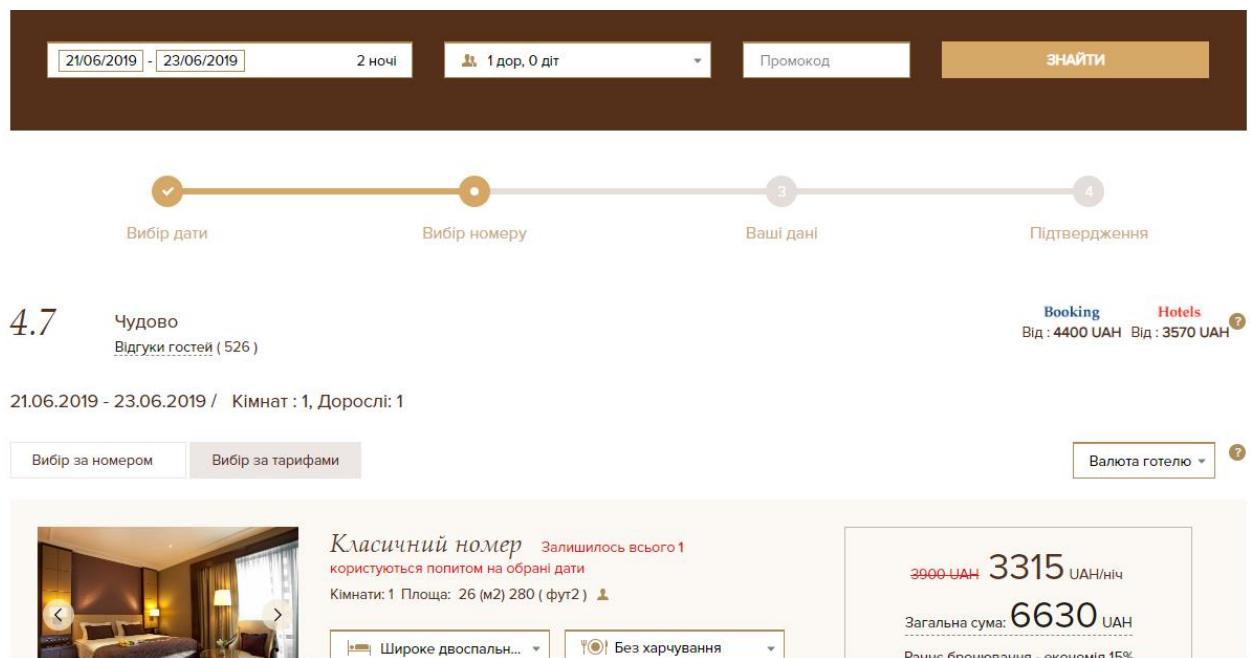


Рисунок 3.11 - Запропоновані номери "Kharkiv-Palace"

Після того як ми розглянули усі веб-ресурси, у таблиці 3.1 запропонована порівняльна характеристика.

Таблиця 3.1 - Порівняльна характеристика веб-ресурсів

Назва програмного продукту	"Надія"	"Кірофф"	"Kharkov-Palace"
Функціональність	Забронювати самостійно номер неможливо, тільки в телефонному	Можна переглянути необхідну інформацію, забронювати	Цей додаток відображає всі вільні номери за нашими критеріями, ми

	режимі з адміністратором.	номер, написати відгук.	можемо залишити свій відгук після реєстрації.
Інтерфейс користувача	Дуже невдало підібрана кольорова гамма, дуже неприємно для очей, номер забронювати неможливо.	Кольорова гамма гарна, та легко можна забронювати номер.	Кольорова гамма підібрана дуже вдало, легко знайти потрібну інформацію та забронювати номер.

Під час аналізу веб-ресурсів аналогів було проаналізовано усі недоліки та переваги цих ресурсів. До переваг можна віднести такі функції, як бронювання номерів, додавання відгуку, можливість увійти в особистий кабінет адміністратору для зміни інформації та приємний інтерфейс для користувачів. Основним недоліком є відсутність можливості особистого бронювання номерів та незрозумілий інтерфейс для користувачів.

Всі вище наведені переваги та недоліки були враховані при розробці моєї рекомендаційної системи.

3.5 Специфікація вимог до задачі

Специфікація вимог - це опис поведінки системи, що розробляється. Специфікація вимог включає всі взаємодії користувача з програмним забезпеченням, тобто функціональні вимоги. У таблиці 3.2 наведено користувачів системи, терміни проекту та основні функції.

Таблиця 3.2 - Перелік користувачів системи, терміни проекту та основні функції

Адміністратор	Користувач системи, який здійснює підтримку системи.
Незареєстрований клієнт	Особа, що використовує систему та яка при бажанні користуватися інтернетом, повинна заповнити форму після наведення камери телефону на QR-код.
Зареєстрований клієнт	Особа, що використовує систему та має можливість бронювання номеру.
Готель	Місце для відпочинку, оснащене усіма умовами для гарного відпочинку.
Браузер	Програмне забезпечення, що дозволяє користувачу переглядати інформацію в інтернеті.
СУБД	Програмне забезпечення, що дозволяє користувачу створити, додати, видалити, зберегти інформацію з баз даних.
Веб-сервіс	Це апаратне забезпечення, що надає користувачу можливість переглядати інформацію та свої ресурси.
Функція "Бронювання номерів"	Дана функція надає можливість користувачу забронювати номер.
Функція "Авторизація користувача"	Дана функція надає можливість користувачу змінювати необхідну інформацію.
Функція "Зв'язок з адміністратором"	Дана функція надає можливість відправити листа адміністратору.

Функція "Редагування стану номера"	Дана функція надає можливість адміністратору змінювати статус номерів.
Функція "Реєстрації"	Дана функція надає можливість незареєстрованому користувачу, зареєструватись в системі.

Рекомендаційна система, що проектується, має такі функції:

- Бронювання номерів;
- Авторизація користувачів;
- Зв'язок з адміністратором;
- Редагування стану номерів;
- Реєстрація;
- Перегляд номерів.

Даний проект розробляється для трьох користувачів: адміністратор, зареєстрований користувач та незареєстрований. Всю взаємодію користувачів з рекомендаційною системою можна показати за допомогою діаграми варіантів використання. Діаграму варіантів використання наведено на рисунку 3.12.

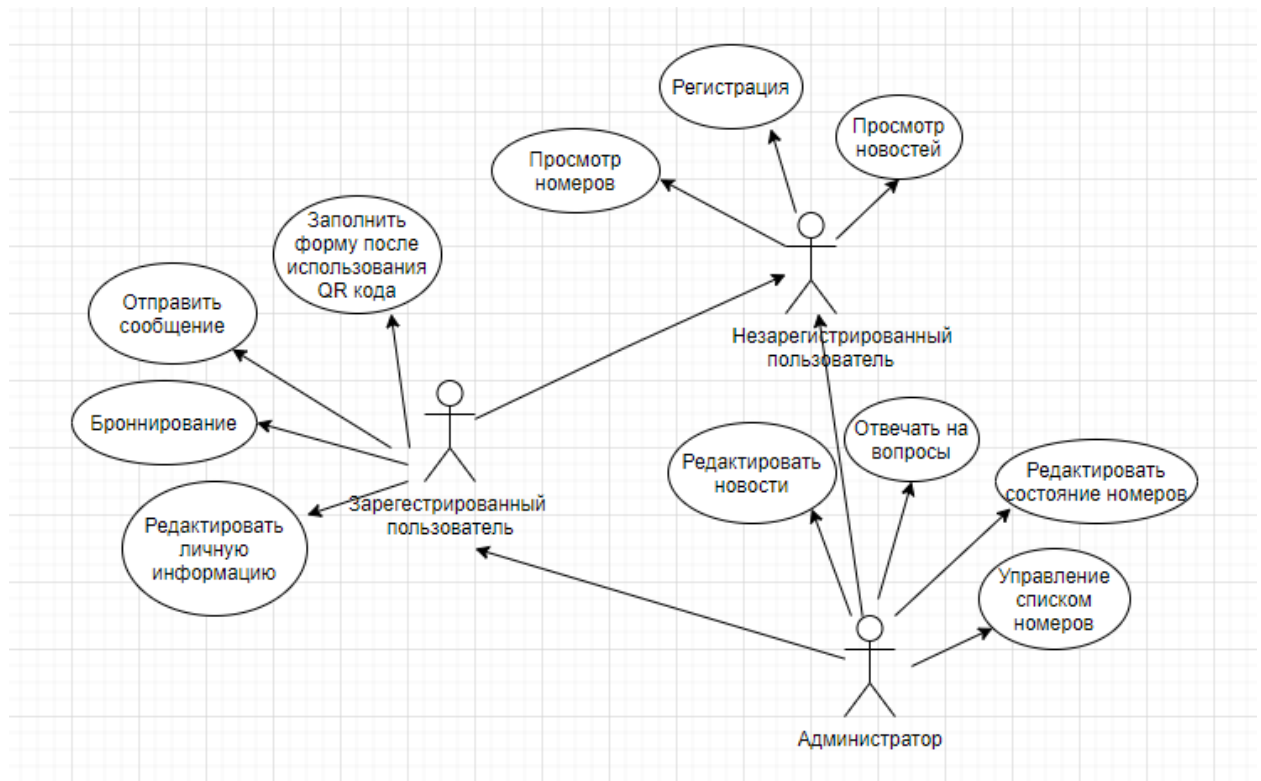


Рисунок 3.12 - Диаграмма вариантов использования

4 ПРОЕКТУВАННЯ СИСТЕМИ

4.1 Опис системних вимог

Основне призначення будь-якої рекомендаційної системи – прогнозування уподобань та надання рекомендацій її кінцевим користувачам. Такі системи можуть використовуватися або бути частиною будь-яких сервісів, що спеціалізуються на наданні користувачам доступу до деякого контенту.

У якості прикладу складної предметної сфери була обрана готельна галузь. Тому створена стратегія рекомендацій буде застосовуватися для веб-сервісу з пошуку номерів готелю. Рекомендаційна система буде однією з обчислювальних частин цієї інформаційної системи, буде повністю в неї інтегрована та буде спеціалізуватися на пропонуванні користувачу номерів, які схожі на ті, що він переглядає у поточний момент.

Сервіс призначений для бронювання номерів в готелі. Виходячи з цілей системи, до неї висуваються наступні системні вимоги.

Система має складатися з клієнтської частини, яка відповідає за отримання даних з сервера для подальшої візуалізації в зручному вигляді для користувача і відправки даних на сервер, а також сервера, який має приймати запити клієнта, обробляти їх, зберігати інформацію про користувацьку активність та прогнозувати клієнтські уподобання, шляхом виконання алгоритмів стратегії рекомендацій, вся інформація, що зберігається повинна бути реалізована у вигляді реляційних баз даних, клієнтська частина інформаційної системи повинна бути реалізована у вигляді веб-додатку, інтерфейс доступу повинен бути зрозумілим і простим для будь якого користувача, управлінська частина для адміністратора повинна повністю

задовольняти вимогам нашої інформаційної системи. Так як такого типу системи отримують велику кількість запитів, то вона повинна бути стійкою до відмов. Система не має бути прив'язана до однієї платформи, повинен бути розроблений програмний інтерфейс додатку.

4.2 Опис функціональних вимог

Основне призначення будь-якої рекомендаційної системи – прогнозування уподобань та надання рекомендацій її кінцевим користувачам. Такі системи можуть використовуватися або бути частиною будь-яких сервісів, що спеціалізуються на наданні користувачам доступу до деякого контенту.

У якості прикладу для розробки рекомендаційної системи була обрана сфера готельної галузі. Тому створена стратегія рекомендацій буде застосовуватися для веб-сервісу з пошуку номерів. Рекомендаційна система буде однією з обчислювальних частин цієї інформаційної системи, буде повністю в неї інтегрована та буде спеціалізуватися на пропонуванні користувачу номерів, які схожі на ті, що він переглядає у поточний момент.

В деякій мірі успіх компанії залежить від умінь виділити, організувати і виконати всі дії з меншими витратами максимально швидко, ніж конкуренти. Таким чином схему діяльності можна назвати невід'ємною частиною будь-якого проекту. Опис дій і бізнес процесів у вигляді тексту зазвичай виходить досить довгим і заплутаним, таким чином використання технологій дозволить змоделювати діяльність організації в чіткому і зрозумілому форматі. Моделі дій і бізнес процесів дозволять нам виділити і організувати інформацію про діяльність організації за допомогою чітких правил їх побудови. Ці моделі можна вважати особливою мовою для формалізації подібної інформації, який містить чітко визначений формат для її подання. Для визначення функціональних вимог

до інформаційної системи розроблена її функціональна модель з використанням стандарту IDEF0. Стандарт IDEF0 - методологія функціонального моделювання (англ. Function modeling) і графічна нотація, призначена для формалізації і опису бізнес-процесів. В IDEF0 розглядаються логічні відносини між роботами. На рисунку 4.1 представлена концептуальна діаграма, яка представляє погляди на функції системи з боку адміністратора.

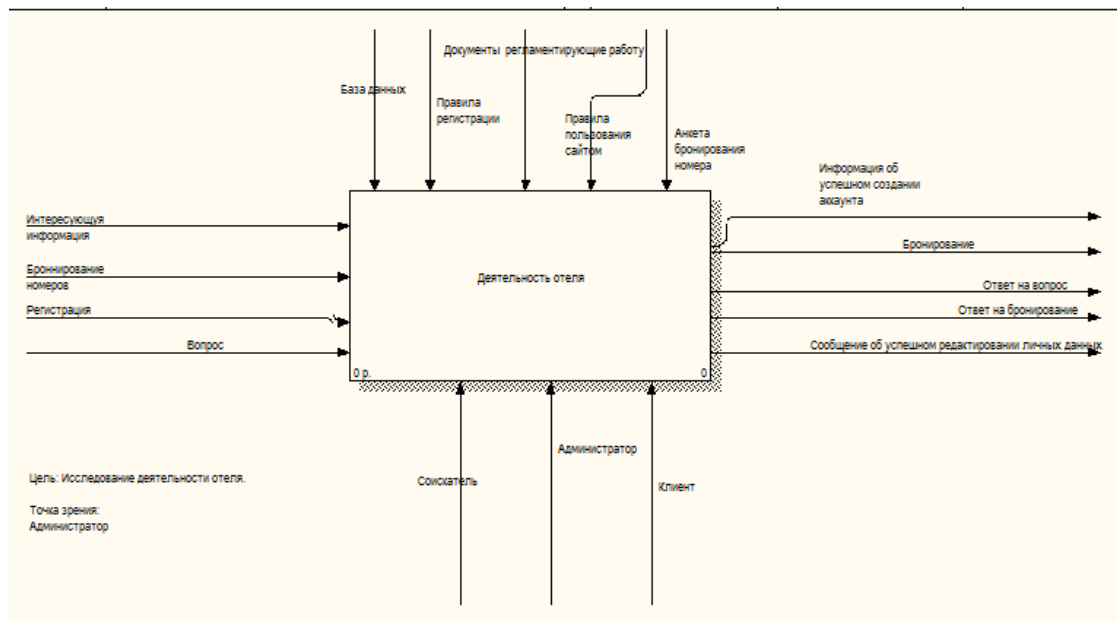


Рисунок 4.1 - Концептуальна діаграма рекомендаційної системи

На діаграмі присутні стрілки входу, виходу, управління та механізму виконання.

- Стрілки входу. Вхід є сировина або інформацію або дані перетворювані функціональним блоком для виробництва виходу.

- Стрілки виходу. Це інформація, що отримується в результаті роботи функціонального блоку.

- Стрілки управління. Ці стрілки відповідають за регулювання того, як і коли виконується функціональний блок.

- Стрілки механізму виконання. Механізми є ресурсом, який виконує дію

що моделюється. За допомогою механізмів виконання можуть моделюватися: ключовий персонал, техніка і обладнання.

Функціональний блок декомпозується, якщо необхідно детально описати його роботу. Декомпозиція блоку дає нам можливість визначити функціональні блоки діаграми. На рисунку 4.2 зображена декомпозиція концептуальної діаграми, яка дозволяє уточнити основні функції рекомендаційної системи і взаємодії між ними.

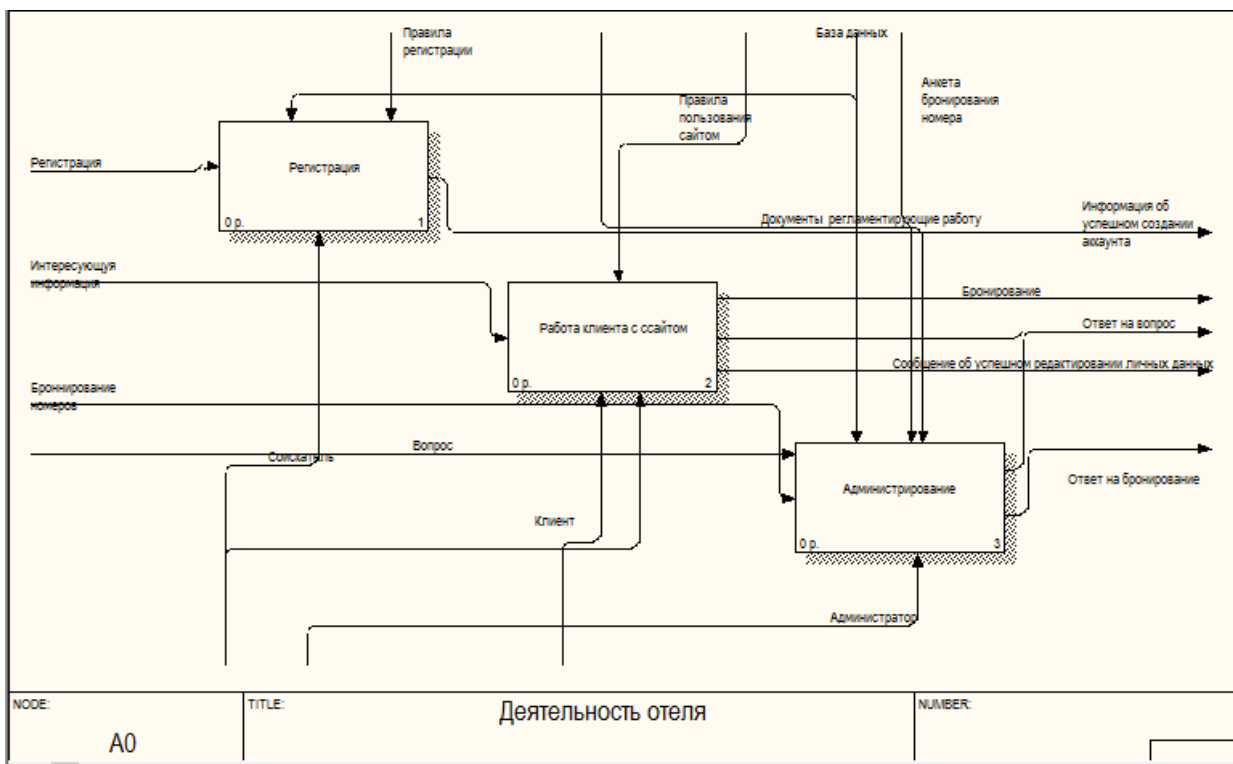


Рисунок 4.2 - Діаграма декомпозиції, що представляє функції системи

На рисунку 4.3 представлена декомпозиція функції «Реєстрація». Ця функція необхідна для відкриття доступу користувачу для відправки замовлення номера і для того, що б клієнт міг переглянути особисту інформацію та сформулювати питання адміністратору.

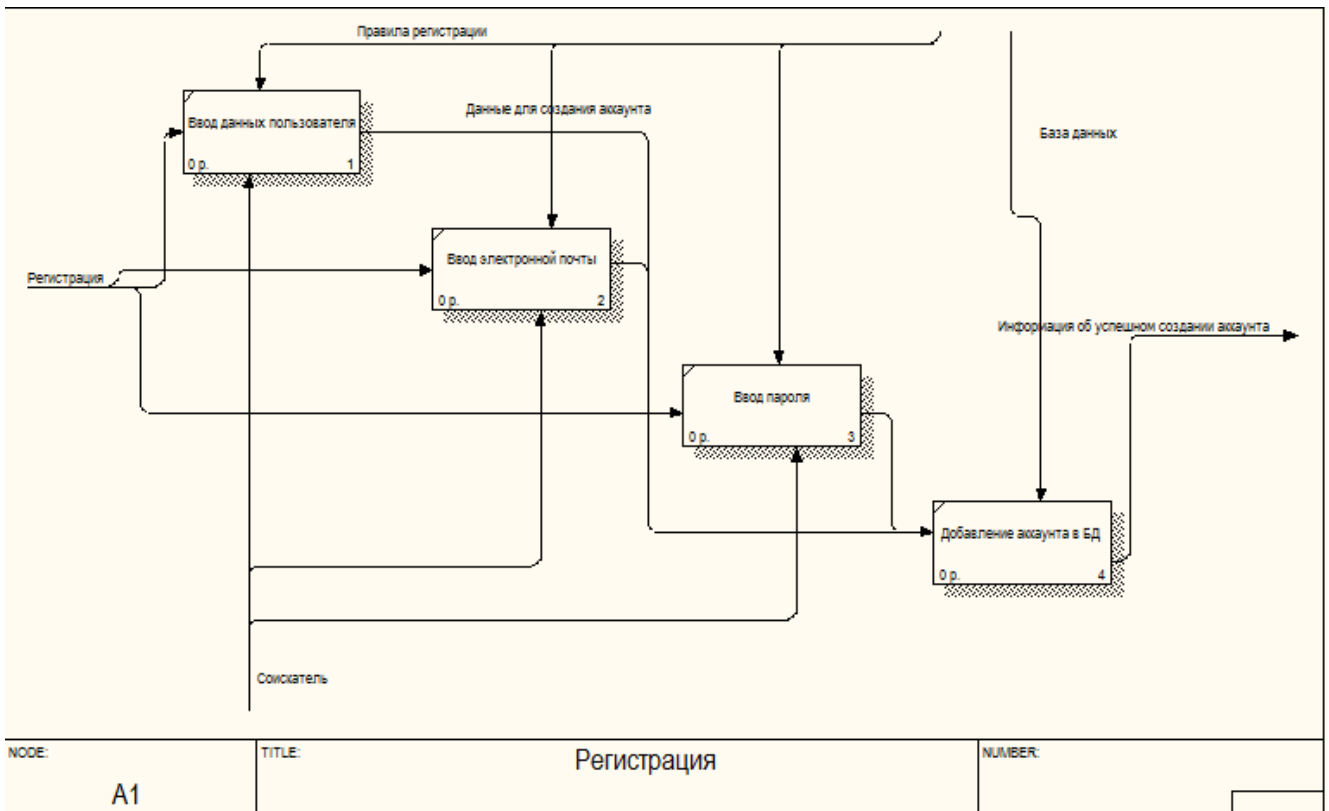


Рисунок 4.3 - Диаграмма декомпозиції функції «Реєстрація»

На рисунку 4.4 представлена декомпозиція функції «Робота клієнта з сайтом». Завданням цієї функції є надання всього необхідного функціоналу для забезпечення зручності клієнтів і користувачів. У діаграмі декомпозицій функції «Робота клієнта з сайтом» представлено замовлення на бронювання, питання адміністратору, перегляд особистих даних, редагування особистих даних. При бронюванні номера користувач повинен ввести свою пошту і ім'я, прізвище, далі замовлення на бронювання номера відправляється адміністратору, потім адміністратор повинен ввести свій логін та пароль і далі адміністратор може редагувати вашу заявку. При перегляді номера користувач вибирає номер за вартістю і на кількість осіб розрахований номер. Так само користувач повинен враховувати правила при бронюванні - це заповнити рядки пошти та імені.

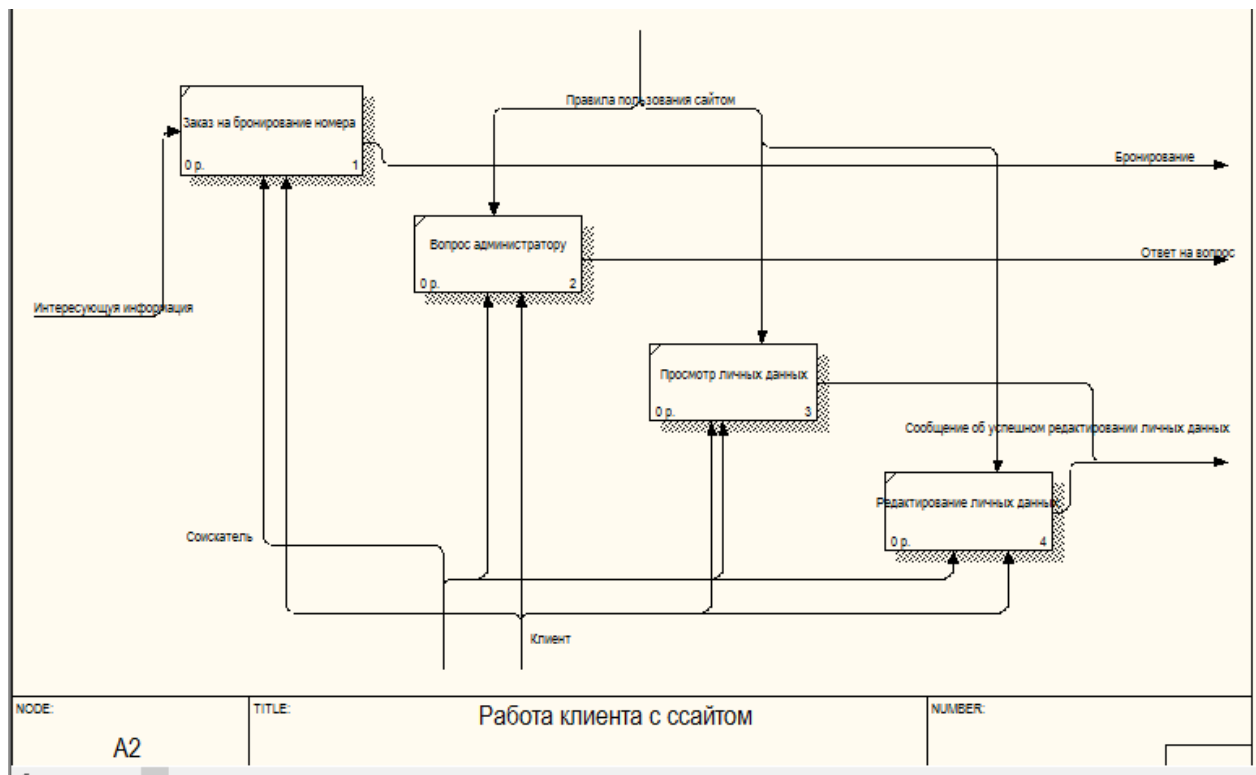


Рисунок 4.4 - Диаграмма декомпозиции функции «Работа пользователя с сайтом»

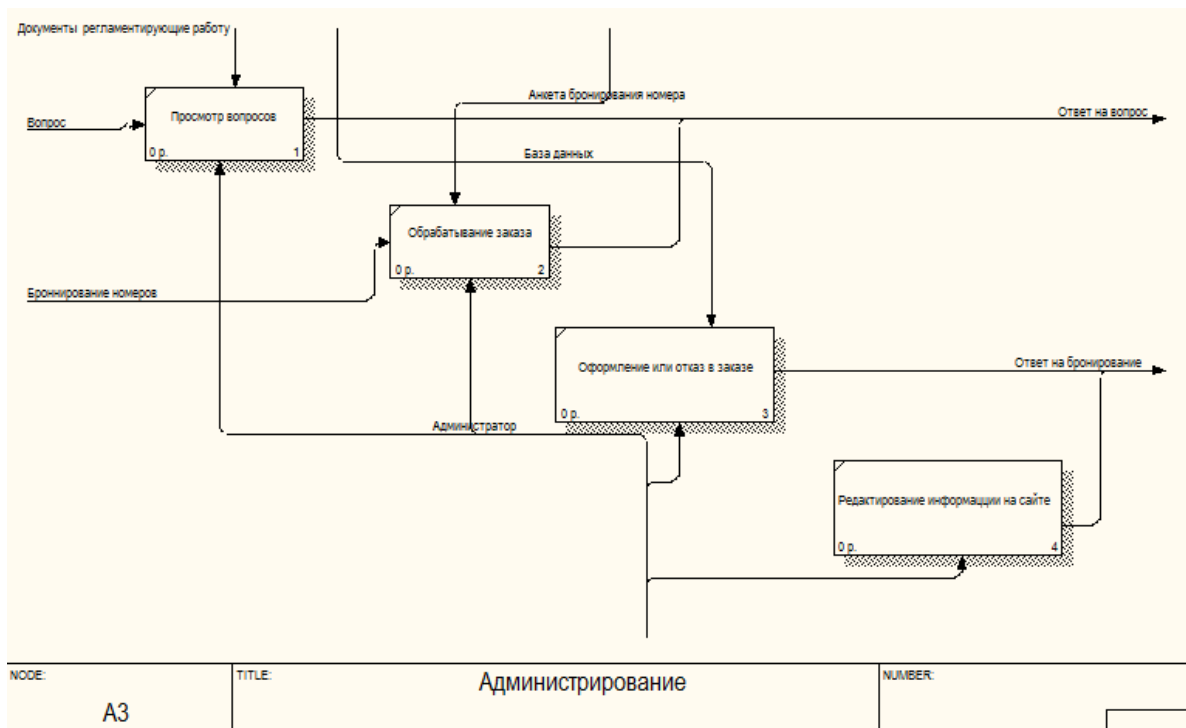


Рисунок 4.5 - Диаграмма декомпозиции функции «Администрирование»

На рисунку 4.5 представлена декомпозиція функції «Адміністрування». Ця функція дає можливість адміністратору керувати інформацією на сайті, і давати відповіді на відправлені резюме і питання.

Сформульована мета моделювання містить питання, на які повинна відповідати модель. Коли стає можливим отримання відповідей на них за допомогою моделі, модель вважається завершеною і задовольняє поставленим вимогам. При побудові декомпозиції першого рівня потрібно стежити за тим, щоб всі блоки на діаграмі лежали всередині певних раніше кордонів моделювання. Перед декомпозицією блоку потрібно впевнитися, чи не призведе це до перевищення установлених раніше глибини деталізації для даної моделі. Ще одне правило полягає в тому, що моделювання IDEF0 має тривати до тих пір, поки стрілки передування (вхід і вихід) переважають на діаграмах.

Таким чином, за допомогою функціонального моделювання вдалося уточнити вимоги до функціональності системи:

- система повинна надавати можливість зареєструватися новому користувачеві;
- система повинна надавати претендентам можливість перегляд вакантних номерів;
- система повинна надавати можливість зареєстрованому користувачу відправляти питання адміністратору і редагувати особисту інформацію;
- система повинна надавати адміністратору спеціальний інтерфейс, для управління роботи сайту;
- система повинна надавати можливість клієнту отримати дані пароля для доступу в інтернет.

4.3 Опис архітектури системи

Відповідно до встановлених системних і функціональних вимог для розроблюваної системи була обрана клієнт-серверна архітектура, при якій

взаємодія модулів відбувається по протоколу HTTP шляхом передачі запитів від клієнта до сервера. У якості архітектурного шаблону сервісу була обрана «MVC» модель.

Шаблон проектування MVC передбачає поділ даних програми, призначеного для користувача інтерфейсу і керуючої логіки на три окремих компоненти: модель, представлення, контролер – таким чином, що модифікація кожного компонента може здійснюватися незалежно [23].

Шаблон проектування MVC передбачає поділ даних програми, призначеного для користувача інтерфейсу і керуючої логіки на три окремих компоненти: модель, представлення, контролер – таким чином, що модифікація кожного компонента може здійснюватися незалежно [23].

Під моделлю, зазвичай розуміється частина, яка містить в собі функціональну бізнес-логіку програми. Модель повинна бути повністю незалежна від інших частин продукту. Модельний шар нічого не повинен знати про елементи дизайну, і яким чином він буде відображатися.

В обов'язки представлення входить відображення даних отриманих від моделі. Однак, представлення не може безпосередньо впливати на модель. Можна сказати, що воно отримує доступ «тільки на читання» до даних.

Представлення має наступні ознаки:

- у представленні реалізується відображення даних, які виходять від моделі будь-яким способом;
- у деяких випадках, представлення може мати код, який реалізує деяку бізнес-логіку.

Прикладами представлення можуть бути: HTML-сторінка, WPF форма, Windows Form.

Контролер інтерпретує дії користувача, сповіщаючи модель про необхідність змін. Контролер забезпечує «зв'язок» між користувачем та системою, він направляє дані від користувача до системи и навпаки, використовує модель и представлення для реалізації необхідної дії.

5 РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Обґрунтування вибору СУБД

MySQL - це найпоширеніша повноцінна серверна СУБД. MySQL дуже функціональна, що вільно розповсюджується СУБД, яка успішно працює з різними сайтами і веб додатками. Навчитися використанню цієї СУБД досить просто, так як на просторах інтернету ви легко знайдете більшу кількість інформації.

Гнучкість MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Ця база даних пропонує досить багато інструментів для розробки додатків. MySQL вважається гарним рішенням для малих і середніх застосувань [24].

Переваги MySQL:

- Простота в роботі - встановити MySQL досить просто. Існують різноманітні програми, наприклад GUI, дозволяє досить легко працювати з БД;
- Багатий функціонал - MySQL підтримує більшість функціоналу SQL;
- Безпека - велика кількість функцій забезпечують безпеку, які підтримується за замовчуванням;
- Не потребує коштів на встановлення, як Oracle;
- В MySQL є процедури, а в PostgreSQL немає, є тільки функції;
- Незважаючи на те, що в MySQL не реалізований весь SQL функціонал, MySQL пропонує досить багато інструментів для розробки додатків.

Недоліки MySQL:

- Відомі обмеження - за задумом в MySQL закладені деякі обмеження

функціонала, які іноді необхідні в особливо вимогливих додатках;

- Проблеми з надійністю - через деяких способів обробки даних MySQL (зв'язку, транзакції, аудити) іноді поступається іншим СУБД по надійності;

Можливості сервера MySQL:

- простота у встановленні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50млн;
- швидкість обробки даних – спрощення деяких стандартів дозволяє MySQL значно збільшити продуктивність у порівнянні з PostgreSQL, який при простих операціях читання може значно уповільнити сервер;
- наявність простої і ефективної системи безпеки;
- масштабованість – MySQL легко працює з великими обсягами даних та легко масштабується.

5.2 Опис розробленої БД

Так як MySQL має реляційний тип баз даних, то потрібно спроектувати схему бази даних, яка повністю буде покривати наші вимоги. Для реляційних баз даних необхідно виконати нормалізацію. Кінцевою метою нормалізації є зменшення потенційної суперечливості інформації, збереженої в базі даних.

Форми нормалізації, які ми будемо застосовувати:

- перша нормальна форма – відношення відповідає першій формі тоді, коли на перетині кожного стовпця і кожного рядка знаходяться тільки елементарні значення атрибутів і не містяться групи, що повторюються;
- друга нормальна форма – відношення знаходиться у другій формі тоді, коли воно знаходиться у першій нормальній формі та кожен описовий атрибут функціонально повно залежить від первинного ключа;

– третя нормальна форма – відношення знаходиться у третій формі тоді, коли воно виконує обмеження другої нормальної форми та усі описові атрибути відношення взаємно незалежні і повністю залежать від первинного ключа, тобто кожний описовий атрибут залежить від ключа[24].

Розроблювана БД буде приведена до 3-ї нормальної форми, що забезпечить такі властивості даних:

- наявність основного ключа: мінімальний набір колонок, які ідентифікують запис;
- уникнення повторень даних, що можуть зустрічатись різну кількість разів в різних записах;
- атомарність даних – кожен атрибут повинен мати лише одне значення, а не множину значень.

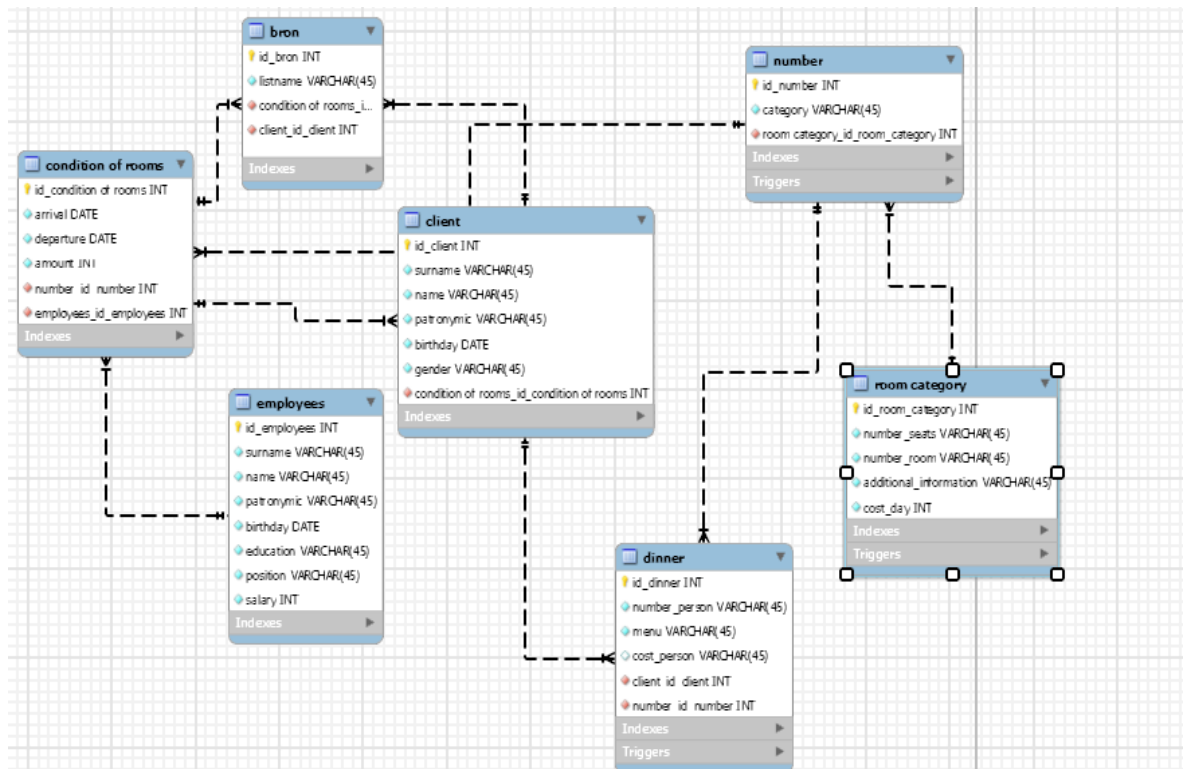


Рисунок 5.1 - Схема бази даних

На рисунку 5.1 зображено 7 таблиць, з них 6 головних: condition of rooms, брон, number, client, dinner, employees і 1 допоміжна: room category. Кожна з таблиць необхідна для того що зберігати необхідні дані.

Таблиця client зберігає дані про клієнта такі як: прізвище, ім'я, по батькові, дата народження, стать, а також користується даними, які зберігаються в допоміжній таблиці брон, для того щоб зберігати дані про те, який номер забронював клієнт.

Таблиця employees зберігає дані про працівників готелю, серед них: прізвище, ім'я, по батькові, дата народження, освіта, посада, заробітна плата, а також користується даними з допоміжної таблиці condition of rooms, яка в свою чергу зберігає такі дані про стан номеру, як: дата виїзду, заїзду та сума .

Таблиця number зберігає дані про всі номери, такі як: категорія, а також користується даними з допоміжної таблиці room category, вона зберігає такі дані як: кількість місць, кількість кімнат, додаткова інформація, вартість за день.

Таблиця dinner зберігає дані про послугу готелю - меню, та зберігає такі як: кількість персон, замовлення з меню та вартість заказу на одну персону.

6 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЕКСПЕРИМЕНТАЛЬНИЙ АНАЛІЗ АЛГОРИТМУСТВОРЕННЯ РЕКОМЕНДАЦІЙ

6.1 Вибір мови програмування

Для розробки серверної частини була обрана мова PHP. Це мова загального призначення з відкритим вихідним кодом. PHP спеціально сконструйований для веб-розробників і його код може впроваджуватися безпосередньо в HTML. На сьогоднішній день PHP є найбільш поширеною мовою веб-програмування. Переважна більшість сайтів і веб-сервісів в інтернеті написані за допомогою PHP.

До переваг PHP відносять наступні:

- для всіх найбільш поширених операційних систем (Windows, MacOS, Linux) є свої версії пакетів розробки на PHP, а це значить, що є можливість створювати веб-сайти на будь-якій з цих операційних систем;
- PHP може працювати у зв'язці з різними веб-серверами: Apache, Nginx, IIS;
- простота і легкість освоєння;
- PHP схожий на мову C, тому, знаючи C або одну з мов з сі-подібним синтаксисом, буде просто опанувати PHP;
- PHP підтримує роботу з великою кількістю систем баз даних (MySQL, MSSQL, Oracle, Postgre, MongoDB та інші);
- поширеність хостингових послуг і їх дешевизна. Як правило, хостингові компанії розміщують веб-сайти на PHP на веб-серверах Apache або

Nginx, які працюють на одній з операційних систем сімейства Linux. І веб-сервери, і операційні системи на базі Linux безкоштовні, що знижує загальну вартість використання хостингу;

- постійний розвиток, PHP продовжує розвиватися, виходять нові версії, які адаптують мову програмування до нових реалій[25].

Серед фреймворків вибір припав на Symfony, який має велике ком'юніті та відмінну документацію. Symfony – вільний PHP фреймворк для швидкої розробки веб-додатків і виконання рутинних завдань веб-програмістів. Розробка і підтримка фреймворку спонсорується французькою компанією Sensio. Symfony складається з набору не пов'язаних між собою компонентів, які можна використовувати повторно в проектах.

Переваги:

- потужна екосистема навколо фреймворка, з хорошим співтовариством і безліччю розробників;
- хороша і постійно оновлювана документація для всіх версій фреймворка;
- безліч різних компонентів для повторного використання;
- пропонує механізм функціональних і модульних тестів для знаходження помилок у веб-додатку;
- підходить для складних і навантажених веб-проектів, електронної комерції.

Недоліки: незважаючи на хорошу документацію, фреймворк є складним для вивчення.

Для розробки клієнтської частини була обрана мова програмування javascript. Це мультипарадигмова мова програмування. Підтримує об'єктно-орієнтований, імперативний і функціональний стилі. JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів

додатків. Найбільш широке застосування знаходить в браузерях як мова сценаріїв для додання інтерактивності веб-сторінок. Основні архітектурні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу [26].

У якості фреймворка для клієнтської частини був обраний React. Це JavaScript-бібліотека з відкритим вихідним кодом для розробки призначених для користувача інтерфейсів. React може використовуватися для розробки односторінкових і мобільних додатків. Його мета – надати високу швидкість, простоту і масштабованість розроблюваним додаткам.

Клієнтська частина використовує API, яке реалізоване на backend, для отримання та відправки даних. Зв'язок відбувається завдяки HTTP-запитам. Використовуються наступні види запитів:

- PUT – для оновлення;
- GET – для отримання;
- DELETE – для видалення;
- POST – для створення;
- PATCH — для часткового оновлення;
- OPTIONS - для опису параметрів з'єднання з цільовим ресурсом.

Аутентифікація відбувається завдяки JSON WebToken (JWT). JWT – це відкритий стандарт (RFC 7519) для створення токенів доступу, заснований на форматі JSON. Як правило, використовується для передачі даних для аутентифікації в клієнт-серверних додатках. Токени створюються сервером, підписуються секретним ключем і передаються клієнту, який в подальшому використовує даний токен для підтвердження своєї особи. Завдяки такому способу спілкування із сервером є можливість використовувати різноманітні платформи для додатку:

- мобільну;
- браузерну;
- десктопну.

Також відсутня прив'язка до backend частини, що дає можливість переписувати додаток, використовуючи інший фреймворк або взагалі іншу мову програмування.

У якості веб сервера виступає Nginx. Це програмне забезпечення, написане для UNIX-систем. Основне призначення – самостійний HTTP-сервер, або, як його використовують частіше, фронтенд для високонавантажених проєктів. Можливе використання NGINX як поштового SMTP / IMAP / POP3-сервера, а також зворотного TCP проксі-сервера.

6.2 Опис алгоритмів інформаційної системи

Нижче наведено опис та схеми роботи деяких основних алгоритмів сервісу.

На рисунку 6.1 зображено повний цикл для бронювання номеру. Для того, щоб забронювати номер, користувач повинен мати свій акаунт. Для цього він повинен авторизуватись, або якщо він не зареєстрований, то йому необхідно створити новий акаунт. Після авторизації він обирає номер та заповнює усі необхідні поля, та тисне на кнопку «Забронювати». Якщо все зроблено вірно, то буде заброньований номер.

На рисунку 6.2 зображено схему алгоритму пошуку номеру.

Щоб знайти необхідний номер, користувач має обрати необхідний йому тип номера та завдати критерії пошуку. Серед наведених об'яв обрати необхідну.

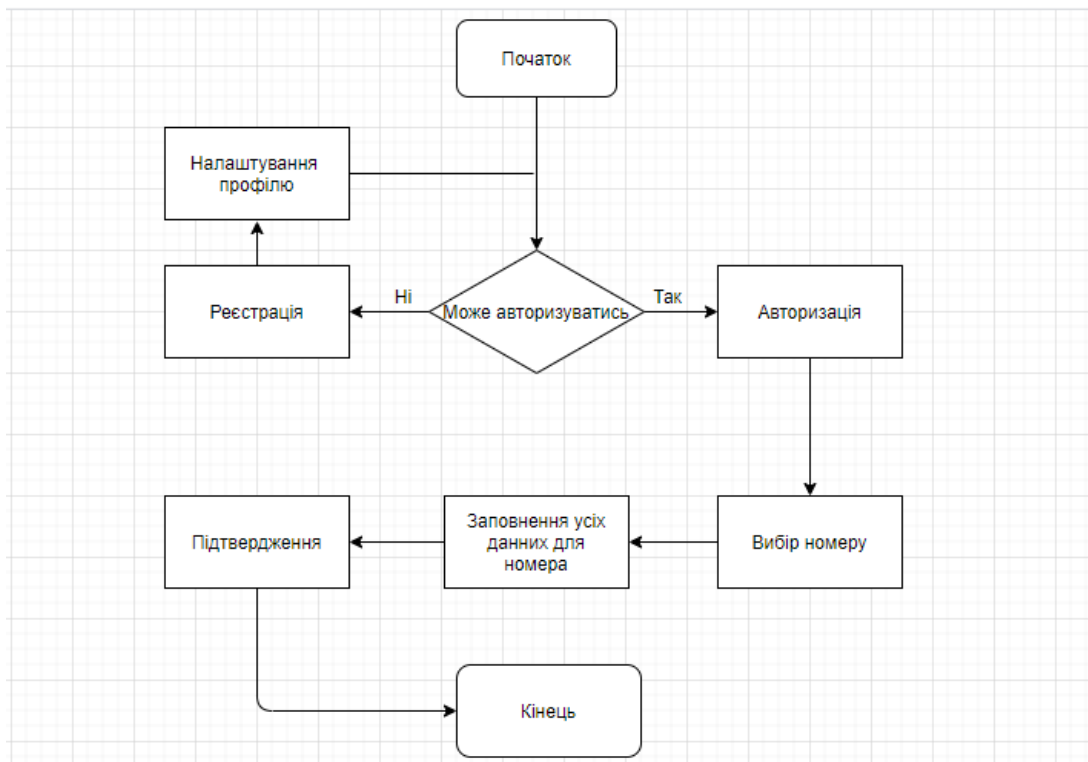


Рисунок 6.1 – Схема алгоритму бронювання номеру

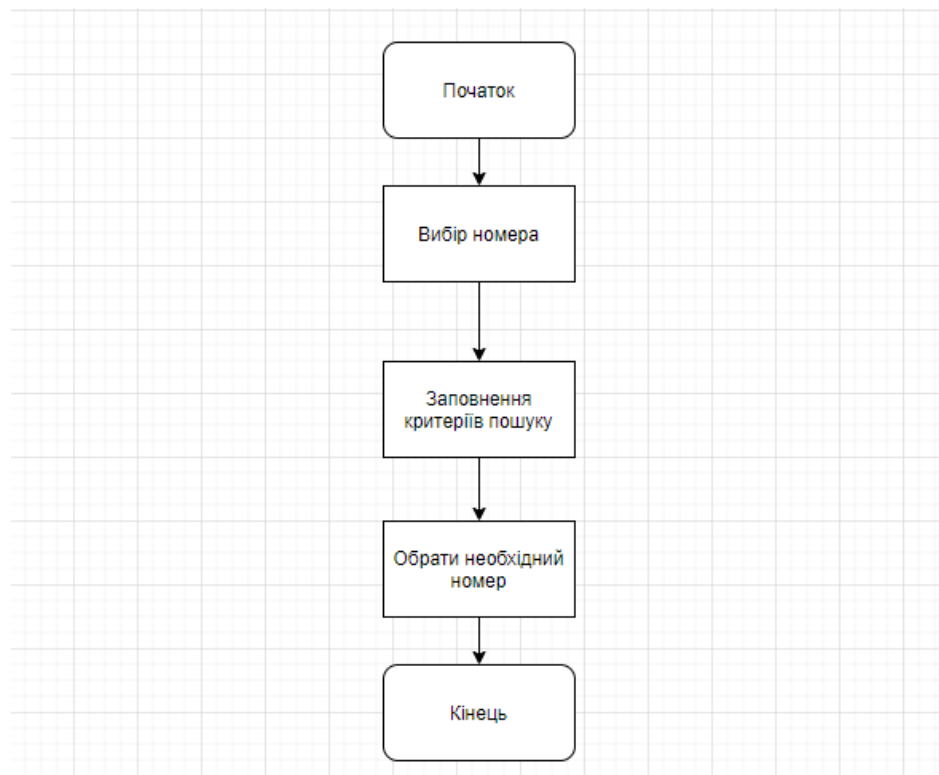


Рисунок 6.2 – Схема алгоритму для пошуку номеру

На рисунку 6.3 зображено спрощену схему роботи рекомендаційної системи (РС).

Для відображення результатів роботи РС користувач має обрати один номер, що пропонує система після виконання пошуку. На початку алгоритм перевіряє кількість номерів, що вже були переглянуті користувачем, за поточну сесію. Результати цієї перевірки можуть призвести до двох наступних варіантів:

- кількість номерів виявляється менше заданої кількості n ;
- кількість номерів виявляється більше заданої кількості n .

При виконанні першого варіанту умови, обране номерів обирається в якості «ідеальної» точки, потім за формулами (2.3) та (2.5) відбувається розрахунок оцінок близькості інших номерів до обраного центру та обирається k переможців.

При виконанні другого варіанту умови, останні n переглянутих номерів (що обираються за методом «ковзаючого вікна») становлять «ідеальну точку», потім за формулами (2.3) та (2.5) відбувається розрахунок оцінок близькості інших номерів до кожного з номерів, які становлять центр, а далі за формулою (2.4) розраховується результуюча оцінка близькості, стосовно обраного центру та обирається k переможців.

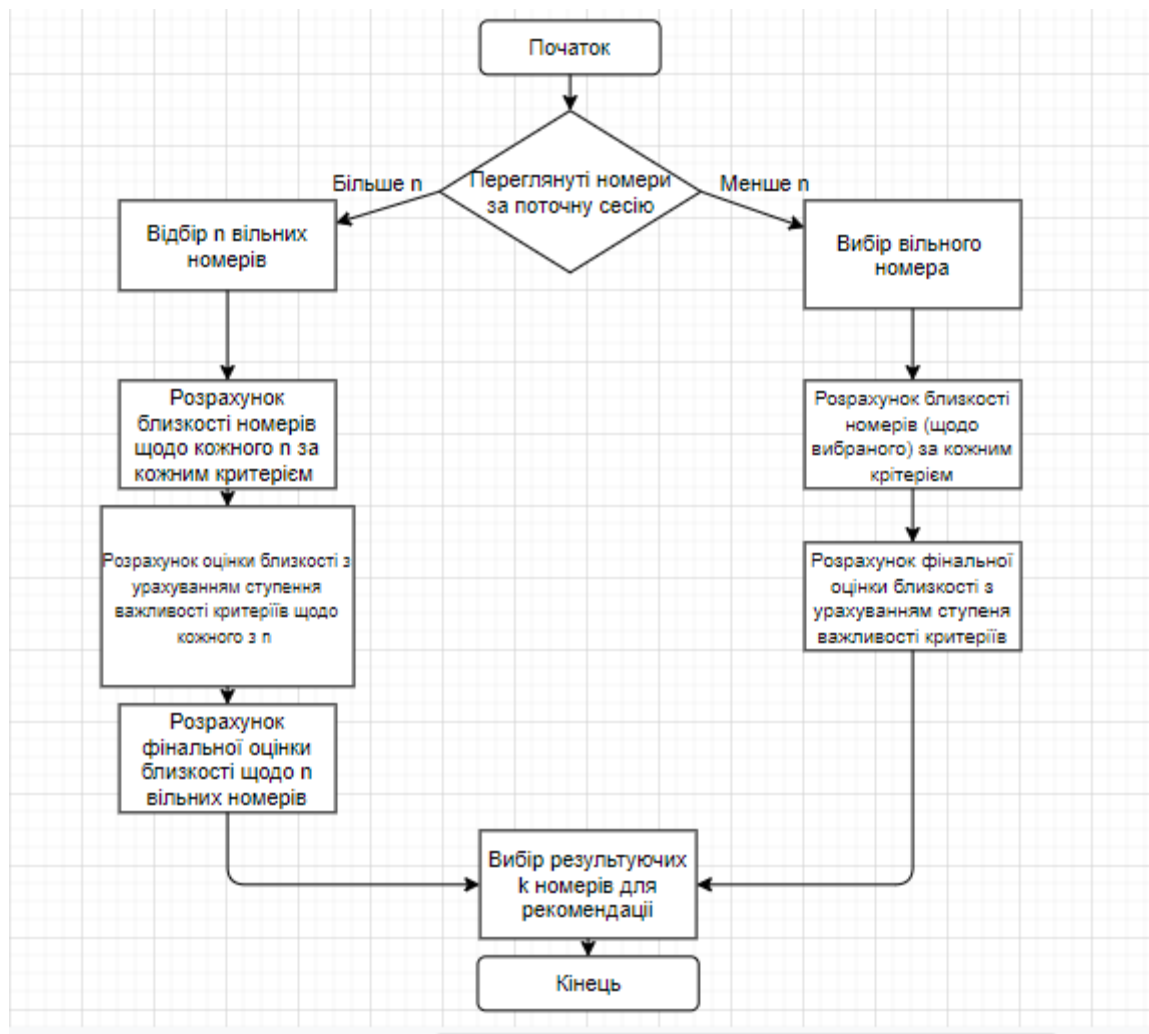


Рисунок 6.3 – Схема алгоритму роботи рекомендаційної системи

6.3 Експериментальний аналіз та наочні результати роботи РС

Експериментальна частина даної роботи полягає у підборі таких параметрів створеної стратегії рекомендацій, які давали б найкращі результати по прогнозуванню уподобань користувача. Результати рекомендацій при цьому порівнюються евристично, тобто шляхом зіставлення різних варіантів кінцевих переможців та визначення того, що найкраще відображає користувацькі потреби.

Підбір параметрів відбувається шляхом аналізу законів функціонування предметної області, переваг середньостатистичного користувача (особиста

думка такого користувача представляється при цьому експертом предметної області), які він надає тим чи іншим критеріям під час пошуку.

Значення n , що дорівнює кількості елементів, які становлять «ковзаюче вікно» дорівнює трьом, проте воно може відрізнятись в залежності від предметної сфери рекомендованих продуктів.

У межах РС для номерів використовуються наступні критерії та їх відносна важливість:

- місто –0.3;
- кількість місць –0.15;
- ціна – 0.20;
- оплата – 0.025;
- харчування –0.03;
- ліжка-0.02;
- дата заїзду –0.25;
- дата виїзду - 0.25;
- передоплата –0.

Значення параметрів для розрахунку близькості за формулою 2.5 для кожного критерію, що представлений у кількісній шкалі описані у таблиці 6.1.

На рисунках 6.4 – 6.8 можна побачити результати роботи рекомендаційної системи. Рисунки 6.5 – 6.6 відображають прогнози щодо одного готелю, тому можна спостерігати невелику різницю у значеннях основних критеріїв, наприклад дату заїзду, дату відїзду та на яку кількість місць розрахований номер. Тобто система рекомендує близькі, «шаблонні» варіанти, що добре відповідають відкритому оголошенню.

На рисунках 6.7–6.8 можна спостерігати значно помітнішу різницю, між прогнозованими варіантами. Це результат застосування принципу «діалогу».

Таблиця 6.1 – Значення параметрів для розрахунку оцінки близькості

Назва критерію, одиниці вимірювання	Назви параметрів		
	Offset	Scale	Decay
Адреса (Геолокація), км	1	5	0,5
К-сть місць, шт	0	1	0,5
Ціна, грн	4% (відмінність від ціни)	15% (відмінність від ціни)	0,5
Оплата	0	2	0,5
Харчування	6%	18%	0,5
Дата від'їзду	5%	12%	0,5
Дата заїзду	5%	12%	0,5
Ліжка	0	1	0,5
Передоплата, грн	5%	15%	0,5

Система поступово «забуває» попередню активність користувача, та запам'ятовує поточну. Так, на цих рисунках бажання користувача змінилися, тому до рекомендації потрапили нові оголошення, що значно менше схожі на його попередню активність, проте також залишилося оголошення, що більше схоже на попередні рекомендації, це означає, що попередня активність, ще не повністю «стерта», або ж користувача цікавлять декілька варіантів, які він шукає паралельно тому різні оголошення і потрапляють до рекомендацій.

Доступные отели в Новом Свете на 20 – 27 декабря 2020



Направление
Отель ГК Новый Свет

Заезд
20 дек. 2020

Выезд
27 дек. 2020

1 номер для
2 гостей

Найти

Рисунок 6.4 – Заполнения критеріів

♥ Отель ГК Новый Свет 2* от 17 490

[Что находится рядом?](#) [Посмотреть цены](#)




Рисунок 6.5 – Активне оголошення

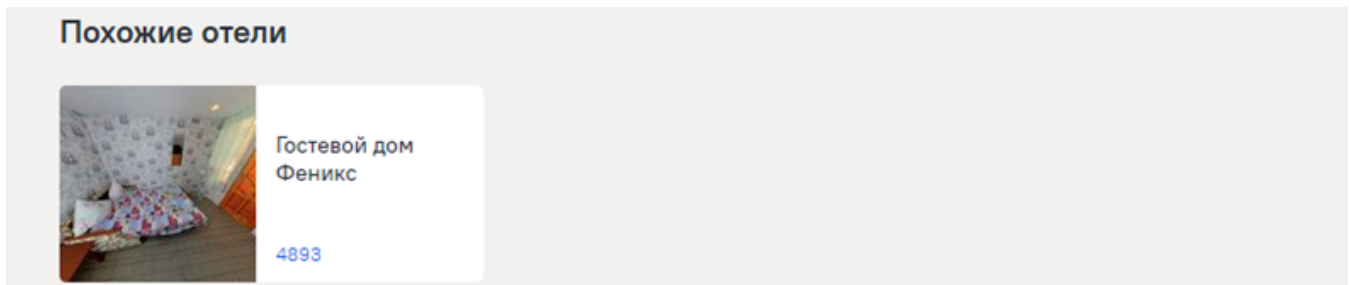


Рисунок 6.6 – Рекомендації до активного оголошення

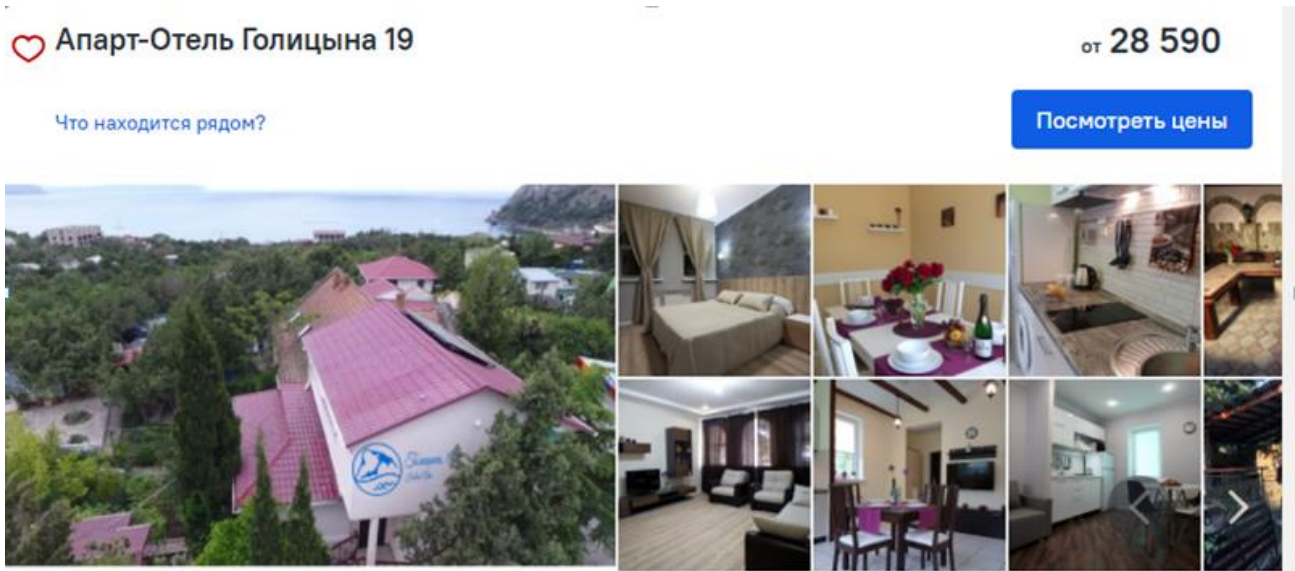


Рисунок 6.7 – Активне оголошення

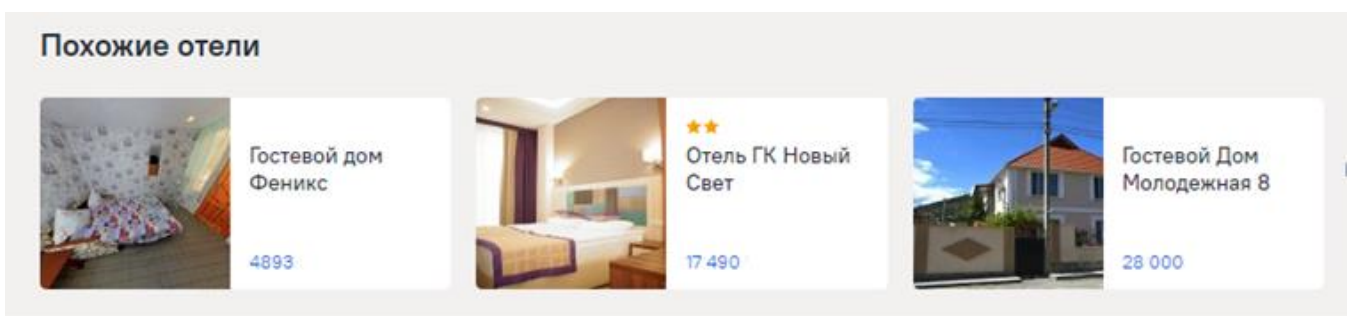


Рисунок 6.8 – Рекомендації до активного оголошення

6.4 Тестування розробленого програмного забезпечення

Тестування програмного забезпечення - це процес, що використовується

для виміру якості розроблюваного програмного забезпечення. Тестування - це процес технічного дослідження, який виконується на вимогу замовників, і призначений для вияву інформації про якість продукту відносно контексту, в якому він має використовуватись. До цього процесу входить виконання програми з метою знайдення помилок.

На сьогоднішній день не існує таких методів тестування, які б могли виявити повністю всі помилки, тому всі існуючі методи діють в рамках перевірки розроблюваного програмного забезпечення. Цей процес називається верифікація. Є багато рівнів, методів та технік тестування, але найбільш доцільні для даної системи буде модульне, інтеграційне та системне тестування.

- Модульне тестування – це таке тестування, в якому перевіряється кожен модуль системи, незалежно від програмного продукту, який розробляється. Його ціллю є показати, що кожен модуль може працювати окремо.

- Інтеграційне тестування – це такий вид тестування коли всі модулі об'єднуються і тоді тестуються разом.

- Системне тестування – це тестування функціональних та не функціональних вимог.

- Тестування безпеки - це тестування, яке використовується для визначення захисту даних. Для того щоб мати можливість змінювати дані на сайті потрібно авторизуватись, що ускладнює можливість несанкціонованого доступу[4,9].

Щоб визначити успішність, програмне забезпечення відповідає іншим верствам споживання, слід проводити навантажувальні тести. Навантажувальні тести моделюють очікуване використання програми шляхом моделювання безлічі користувачів, які працюють з програмою одночасно.

Якщо модель витримує високі навантаження (наприклад, тисячі користувачів одночасно відвідують веб-сайт), необхідно мати кілька комп'ютерів, щоб імітувати достатнє завантаження. Для досягнення цієї мети

можна налаштувати групу комп'ютерів для створення інтенсивного навантаження, моделюється тестами. Це група комп'ютерів складається з одного або декількох контролерів тестування і одного або декількох агентів тестування. Агенти тестування виконують тести і можуть створювати систему навантаження, що моделюється.

Модульні тести використовуються в навантажувальні тести для перевірки роботи сервера за допомогою API. Як правило, вони призначені для серверів, доступ до яких здійснюється не через браузер, а через "товсті" клієнти та інші служби сервера. В якості одного з прикладів можна назвати додаток з інтерфейсом Winforms. В даному випадку розробляються модульні тести. Іншим прикладом є інший сервер, який викликає перший сервер за допомогою веб-служб. Крім того, можливі випадки, коли дворівневі клієнти безпосередньо викликають SQL. Тоді можна розробити модульні тести для безпосереднього виклику SQL.

Для того, щоб користуватись сайтом потрібно мати браузер. Веб-сайт було перевірено на сумісність з таких браузерів:

- Google Chrome;
- Mozilla Firefox;
- Internet Explorer;
- Microsoft Edge;

У всіх браузерах система поводитьсь однаково, без дефектів.

Були протестовані дві сторінки мого додатку:


- а) Форма вводу логіна для реєстрації зображена на рисунку 6.9.

Войти или [Зарегистрироваться](#)

[Напомнить пароль](#)

Рисунок 6.9 - Форма вводу логіна для входу

б) Форма вводу даних для бронювання номеру на рисунку 6.10

 **Контактные данные**
Пожалуйста, заполните все поля

[Указать имена всех гостей](#)

Рисунок 6.10 - Форма вводу своїх даних для бронювання номеру в рекомендаційній системі

Було протестовані дві сторінки рекомендаційної системи, все працює коректно, дані користувачів заносяться в систему та в подальшому використанні нашої системи не будуть виникати питання з реєстрацією при вході в рекомендаційну систему та вводу даних при бронюванні номеру.

Також на рисунку 6.6 можна побачити результат прогнозу щодо одного готелю, була невелика різниця у значеннях основних критеріїв, наприклад дату заїзду, дату від'їзду та на яку кількість місць розрахований номер. Система рекомендувала близькі, «шаблонні» варіанти, що добре відповідають відкритому оголошенню, за поточну сесію був переглянутий один номер, тому система порекомендувала один варіант, рекомендаційна система працює також коректно.

На останній протестованій сторінці, яку можна побачити на рисунку 6.8, застосовано принцип "діалогу", тому ми можемо спостерігати значну різницю між прогнозованими варіантами. Система поступово «забуває» попередню активність користувача, та запам'ятовує поточну. Так, бажання користувача змінилися, тому до рекомендації потрапили два нових оголошення, що значно менше схожі на його попередню активність, проте також залишилося одне оголошення, що більше схоже на попередні рекомендації, це означає, що попередня активність, ще не повністю «стерта», або ж користувача цікавлять декілька варіантів, які він шукає паралельно тому різні оголошення і потрапляють до рекомендацій.

Система в усіх випадках працює коректно.

ВИСНОВКИ

В ході аналізу сучасного стану досліджуваної проблеми, пов'язаної з використанням рекомендаційних систем, були розглянуті основні підходи до їх створення, що використовуються сучасними інтернет - платформами. Стало зрозуміло, що проблема створення рекомендаційних систем зі складними предметними сферами є дуже актуальною, бо жодна з існуючих і добре досліджених стратегій не підходить для них, адже вони потребують знань про уподобання окремо взятого користувача чи групи користувачів. Для сервісів, які мають високу частоту оновлення елементів та/або низьку частоту їх перегляду, створення такої бази знань, на основі якої можна було б давати релевантні рекомендації, є складною задачею, бо такі особливості функціонування подібних сервісів тягнуть за собою не прогнозованість уподобань користувачів та велику розрідженість матриць оцінок елементів.

Після аналізу основних стратегій по створенню рекомендацій на предмет їх застосування для вирішення поставленої задачі, були виділені їх сильні та слабкі сторони, а також відібрані аспекти та ідеї, на основі яких була розроблена стратегія рекомендацій, яка підходить для рекомендування елементів зі складних предметних сфер. Ця стратегія базується на контентних рекомендаціях та рекомендаціях на основі знань. Вона створює рекомендації на основі лише опису товарів, слідкуючи при цьому, за активністю користувачів та поступово змінюючи вектор рекомендацій, якщо уподобання користувача змінюються за час перегляду, при цьому стратегія не передбачає збереження інформації про уподобання користувача за попередні проміжки часу. Це зроблено через особливості предметної області, бо низька частота переглядів не гарантує створення релевантної бази знань, адже існує велика імовірність того, що уподобання користувачів будуть відрізнятися при кожному такому перегляді.

У якості прикладу складної предметної сфери була обрана сфера готельної галузі. Тому у якості об'єкту дослідження були обрані системи, що спеціалізуються на пошуку нерухомості, був проведений їх аналіз та виділені їх основні можливості та принципи функціонування. Була вирішена задача проектування інформаційного забезпечення однієї з таких систем та виконана програмна реалізація додатку, що спеціалізується на пошуку номерів з використанням, реалізацією та впровадженням розробленої стратегії по створенню рекомендацій. Наочні результати роботи розробленої рекомендаційної системи, показали, що вона здатна створювати релевантні рекомендації і може використовуватися для різноманітних сервісів, що охоплюють складні предметні сфери.

Створена рекомендаційна система може бути застосована в сфері із схожою тематикою.

Перевагою даного продукту є простий і зручний інтерфейс для звичайного користувача так і для адміністратора.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. JannachD., ZankerM., FelfernigA., FriedrichG.Recommender Systems: AnIntroduction. CambridgeUniversityPress(2010).
2. P. Lops, M. Gemmis, G. Semeraro Recommender Systems Handbook Springer-Verlag,2011.
3. C. C. Aggarwal, Recommender Systems: The Textbook. Springer. C: 8-37,2016
4. Introduction to recommender systems [Електронний ресурс]. Режим доступу: <https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>
5. K. Falk, Practical Recommender Systems 1st Edition. Manning Publications C:150-174,2019
6. Recommender Systemsin Practice [Електронний ресурс]. Режим доступу: <https://towardsdatascience.com/recommender-systems-in-practice-cef9033bb23a>
7. Xiaoyuan Su, Khoshgoftaar T. M. A Survey of Collaborative Filtering Techniques. Advancesin Artificial Intelligence,2009.
8. U., K., Viswanathan. Optimization of User Based Collaborative Filtering. LAP LAMBERT Academic Publishing. C:7-22,2019
9. S., Berkovsky, I., Cantador, Domonkos T. Collaborative Recommendations: Algorithms, Practical Challenges and Applications. World Scientific Publishing Co,2019
10. B., Jesús; O., Fernando; A.; Bernal, A collaborative filtering approach to mitigate the new user cold start problem.2012
11. Wikipedia [Електронний ресурс]. Coldstart (Computing). Режим доступу:[https://en.wikipedia.org/wiki/Cold_start_\(computing\)](https://en.wikipedia.org/wiki/Cold_start_(computing)).

12. Wikipedia [Электронный ресурс]. Knowledge-basedrecommendersystems. Режим доступа: https://en.wikipedia.org/wiki/Knowledge-based_recommender_system.
13. Р. Берк, Системы рекомендаций, основанные на знаниях, Энциклопедия библиотечных и информационных наук. С: 180-200,2000.
14. Y. Salem, J. Hong, W. Liu. History-Guided Conversational Recommendation, WorldWideWeb (WWW),2014.
15. А. Фелфернигу и Р. Бёрку, Системы рекомендаций на основе ограничений: Технологии и проблемы исследований. АСМ Международная конференция по электронной торговле. С: 17-26,2008.
16. Л. Чен и П. Пу. Рекомендации на основе критики: обзор и новые тенденции. «Моделирование пользователей и адаптированное к пользователю взаимодействие» (UMUAI), С: 125-150,2012.
17. Salter,J.;Antonopoulos,N.(January2006).CinemaScreenRecommenderAgent: CombiningCollaborativeandContent-BasedFiltering. IEEE Intelligent Systems.2006
18. T., Chen, W., LiHan, H., D., Wang, Y., X., Zhou; B., X., B., Y., Zang, Content Recommendation System Basedon Private Dynamic User Profile. International Conference on Machine Learning and Cybernetics,2014.
19. Window ingmeyhod. [Электроний ресурс]. Режим доступа: <https://wiki.loginom.ru/articles/windowing-method.html>
20. Метод взвешенной суммы критериев в анализе много критериальных решений.[Электроний ресурс]. Режим доступа: <https://bijournal.hse.ru/data/2013/10/11/1281327437/5>.
21. Маклаков С.В. Моделирование бизнес-процессов с ВРwin 4.0 – М.: Диалогмифи, 2002. – 151с.
22. Черемных С.В., Семёнов И.О., Ручкин В.С. Структурный анализ систем: IDEF – технологии – М.:Финансы и статистика, 2003. – 208с.

23. Wikipedia [Электронный ресурс]. Model-View-Controller. Режим доступа:<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>.

24. P. DuBois, MySQLCookbook: Solutions for Database Developers and Administrators. O'ReillyMedia; 3 edition, 2012 – С.866.

25. М. Зандстра PHP: Objects, Patterns, andPractice. «Ozon», 3-издание, 2011. – С.576.

26. Wikipedia [Электронный ресурс]. JS. Режим доступа: <https://en.wikipedia.org/wiki/JavaScript>.

27. Nechyporenko, A. Intelligent information system of heterogeneous medical data analysis / Yerokhin, A., Turuta, O., Babii, A., Nechyporenko, A. // Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2017. – 2017. – Vol. 8098798.– P. 332–335.